# A framework for statistical software development, maintenance, and publishing within an open-access business model

Patrick Wessa

e-mail: patrick@wessa.net

## Summary

There are several fundamental problems with statistical software development in the academic community. In addition, the development and dissemination of academic software will become increasingly difficult due to a variety of reasons. To solve these problems, a new framework for statistical software development, maintenance, and publishing is proposed: it is based on the paradigm that academic and commercial software should be both cost-effectively created, maintained and published with Marketing Principles in mind. The framework has been seamlessly integrated into a highly successful website (www.wessa.net) that operates as a provider of free web-based statistical software.
Finally it is explained how the R framework provides a platform for the development of a true compendium publishing system.

**Keywords:** R, Statistical Software, Compendium, Business Model

# Motivation

Does it matter whether statistical software products are created within a commercial or an academic environment? Yes, we often observe substantial differences between them. On the other hand, there is no reason why we should treat academic software any differently from commercial software. In my view, both types should:

- comply to the same quality standards;
- be easy to use;
- be cost-effectively produced, maintained, and implemented;
- be distributed using sound marketing principles;
- offer a real, measurable "added value".

Economic and Business Theory provides several interesting models that explain the fundamental problems with (statistical) open source software in the academic community. Also, there are several reasons why profit-driven developers may be more efficient than the "internet talent pool" of open source communities and produce software that is more complete and user-friendly (Johnson, 2001).

The development and dissemination of academic software/techniques is increasingly difficult due to a variety of cultural, technical, and economical reasons. This is related to the incentive schemes and dynamic relationships between maintainers of software, and the user-developers (Edwards, 2005). Another reason lies in the impact of the licensing scheme (open versus closed) which has been found to be substantial and persistent - even though the open source and free software community has been found to have a positive impact on the entire industry in terms of standardization (Välimäki, 2005).

# Utility functions

In addition to what Business Theories predict about open source software development, I could add a few remarks based on Micro Economics and personal experience. In Micro Economics the utility function plays an important role because it explains how agents (consumers, producers, etc...) make decisions: they behave rationally and are assumed to maximize their utility, given that resources are scarce and the fact that the world contains various types of uncertainties. A brief exploration of the utility functions of academic software producers, and their universities may provide useful information about how various incentives play a decisive role in the proposed Business Model.

My first observation is that many open source initiatives originate within an academic context: research funds are used to create innovative software and to produce articles that contribute towards ones academic record (which is an important factor in the researcher's utility function). Often, the software product does not count in itself, and is merely considered to be a by-product. This is problematic because it effectively discourages software production and maintenance. If universities are paying license fees for commercial software products then there shouldn't be any reason not to value academic software production. If we cease to treat academic software differently from commercial software then we automatically attribute value to its development in the form of "academic credit" and hereby provide an incentive to the academic developer.

My second observation is that research projects that result in the creation of academic software often neglect to budget sufficient resources for maintenance. Again, the reason why this happens is quite simple: there is no incentive to plan for maintenance because it does not count. Maintenance of academic software does not produce new articles unless the update is substantial and involves new and publishable features. However, if the academic software proves to be useful for educational purposes then the university has a strong incentive to support its creation and maintenance because education is a main factor in its utility function.

Third, the dissemination of research (and the accompanying software) only involves some form of academic publication. This however, does not guarantee that the academic software will be actually used by others. The university's utility is a function of the so-called Impact of Research which is closely related to reputation. Reputation is important because it allows the institution to attract more students, better professors, and more funding through a variety of ways. Therefore it makes sense to stop using the number of publications as the sole measure of academic impact. There are other measures of impact such as citations and a wide variety of usage indicators that might be considered. The clear advantage of usage-based measures is that it equally applies to articles, books, databases, and software. In addition, they are operational and allow any citable item that helps in the creation of new research to be valued and accredited.

If we agree that dissemination should lead to impact then it makes sense to promote the innovations in statistical computing by the use of Marketing Principles. The well-known 4P Marketing Mix Model (Product, Price, Place, Promotion) has been heavily criticized in literature - especially within the context of services and web-related activities - because it is argued that the traditional Marketing Mix paradigm is incompatible with web-based activities. However, several new models (such as the '4S Web-Marketing Mix Model') have been proposed (Constantinides Efthymios, 2002) to account for critical factors that are related to strategy, organization, integration, and technical issues. The impact-based utility function of universities makes the use of Marketing Principles a necessity in any Business Model that is considered.

An important problem arises from the fact that the traditional dissemination of research (articles, books) is very expensive, especially when the traditional impact measures are used. An example may illustrate why this issue is important. Let us suppose, that the creation of a traditional article (published in a closed-access journal) represents a (conservative) labor cost of 15000 €/article. Also, suppose that the cost of dissemination (Rubinstein, 2002) amounts to another 15000 €/article. According to the statistics of Thomson Scientific (2005) the citation frequency (in 2005) of mathematics articles published in 2004 is 0.06 citations/article. The estimated cost per citation (= unit of Impact) can now be computed as (15000+15000)/0.06 = 500000 €/citation. This example illustrates that impact (and hence reputation) comes at a high price for the university. Therefore, there is every reason to search for alternative publishing models that promise to reduce costs and increase the probability of obtaining impact.

From the above we may conclude that the economic utility of academic institutions mainly depends on impact of scientific output (which includes software) and the ability to include academic results (such as software or databases) in education. Impact leads to reputation, and reputation helps to obtain funding in various ways. The utility function of the researcher also depends on (personal) impact which should be adequately measured (based on usage-based measures or citations) in order to provide the incentive to develop and maintain non-article output such as software and databases. Admittedly, many scientists do not cite the software and databases that are used in their research. However, this does in no way imply that we should abandon the valuable concept that research output should be reproducible. Ultimately, we have to change our attitudes towards citing software and databases because it effectively helps to provide incentives to create and maintain them at a high level of quality.

# Business Model

The framework that is presented in this paper is based on the paradigm that academic software should be managed as if it were a commercial product. The observation that most academic software projects are free of charge, does not imply that the process of academic software development (including maintenance and publishing) is not bound to the basic laws of economics - applying to both costs and benefits. Therefore, I would like to propose (some) elements of a business model that allow for the creation, maintenance, and publication of statistical software that is offered within an open-access philosophy. For this purpose I discuss the four components of the Business Model Ontology proposed by Osterwalder (2004): Product, Infrastructure Management, Customer Interface, and Financial Aspects.

### Customers

Some statistical software producers can afford to build a long-term relationship with institutes of higher education through a programme that allows students (and faculty) to benefit from cheap software licenses. This "Push" technique may become an integral part of their Marketing Strategy because it ensures that future generations of decision-makers and software consumers are familiar with their products. This effectively creates a demand for their software products (called "Pull"), making the strategy (very) profitable on the long run. One could argue that there is nothing wrong with this type of commercial relationship because the delivery of user-friendly software may be greatly beneficial to students.

The demand for statistical software in academic research may be partially satisfied by commercial products and services. In many cases however, the commercial solutions are inadequate and this inevitably leads to the creation of specific-purpose academic software packages. As explained before, the academic software that was created to conduct the research is often seen as a by-product and almost never readily available for the purpose of reproducing the results of research.

As a consequence it is here that we have to build a bridge between the authors of statistical research software and the vast community of statistical computing consumers (including students). This bridge is called R framework and has been seamlessly integrated into a website (www.wessa.net) which is part of the successful Xycoon Project (Borghers, Wessa, 2005) that makes statistical computing freely accessible through "Web-Enabled Scientific Services and Applications" for educational and research purposes. The words "freely accessible" are related to the consumers of statistical computing and every aspect that allows them to make good use of the underlying software.

Just like large commercial software producers build a long-term relationship with their customer base, it makes a lot of sense to provide students access to newly developed techniques in the form of academic software. If this is done consistently and in a standardized (cheap) fashion then the university has – as explained before – every reason to value this educational contribution. The proposed Framework will actually help us in this respect. As an extra benefit, it helps scientists to promote the use of newly developed methods on the short run (via Search Engines) and on the long run (today's students are future consumers of statistical computation). As a consequence it makes sense to offer Customers a Product (c.q. website) that is distributed by the use of Internet technology which is relatively cheap and available to a quickly growing number of potential Customers.

**Product**

Developers of academic statistical software are generally not concerned with issues of accessibility. They often have no incentives to worry about it and there seems to be no mechanism that requires them to provide adequate support, and maintenance on the long run.

Within the lines of the business model paradigm, accessibility can only be achieved if certain conditions are satisfied:

- the solutions that are offered by the website must be findable by anyone with internet access ("findability");

- the software should be easy to use by visitors of the website ("usability");
- access to the website should be open and free of charge for (non-commercial) purposes such as education, and science ("open-access");
- technical (hardware and software) requirements to use the software should be limited as much as possible;
- knowledge of programming techniques and mathematical statistics should not be required to make use of the software;
- the ability to serve requests from growing numbers of users must be guaranteed ("scalability").

The R Framework allows us to create, maintain, and publish statistical software (called R module) that participates in the business model. Let us consider an example to illustrate the steps that are involved in creating an R module. Suppose we would like to create a module that allows a visitor to compute a Bivariate Kernel Density plot as described in Lucy et al. (2002) and implemented in the GenKern package as available in CRAN, R Development Core Team (2006).

In the first step, the R module creator needs to define the input of datasets and parameters. In the R module creation form (coded in html) we can specify that the module requires two dataseries to be entered by the user. The software that we want to create is based on the KernSur function of the GenKern package and requires the user to input several parameters. Therefore we define all parameters that are necessary to call the function (including defaults, labels, and descriptions).

In the second step, we write the R code that computes the Bivariate Kernel Density plot based on the datasets and input parameters. The input parameters are always named par1, par2, par3 ... and require type conversion in the R code such as: *par1 <- as(par1,"numeric")*. Obviously the conversion depends on the syntax that is required to invoke the R function. When all parameters have been converted, it is generally a good idea to validate input in order to make sure that the parameters are within reasonable bounds. The actual computation can now be coded: in this example it only involves a single function call. The resulting output that is generated by the R engine, needs to be converted to HTML code and images (postscript and png). There are several functions available to accomplish this as can be seen in the complete R code for this example (Wessa, 2007).

The third step requires us to define meta data about the R module: a detailed description, keywords, titles, hyperlink captions, citations of the literature, etc... When this information has been submitted, the R framework automatically creates a web-based user interface and underlying code that can be used for testing purposes. The R module creator and the editor are both able to review the module and make changes where necessary. It is the responsibility of the editor to finally publish the module, making it visible to search engines and visitors.

One of the reasons why the R framework is very cost-effective is because it creates the actual code (and user interface) that converts user requests to requests for the R engine. The process of code creation is based on meta data that is specified by the module creator and the editor. The meta data is generic, independent of technology, and extendable. This allows us to change the layout or mechanism of the user interface without rewriting R code. The R modules are instantly recreated based on the available meta data and according to the specifications of the user interface. The R modules that are created by the framework are increasingly user-friendly ("usability") and well-suited for marketing purposes. The user feedback that we received in the past 11 months (through an online feedback form) has been overwhelmingly positive. Moreover, the modules are safe to use (they don't require any download or installation) and are compatible with any HTTP-enabled browser.

Most R modules have top 10 search ranks for relevant search phrases in major search engines. This effectively promotes the use of statistical science ("findability") that would otherwise only be available to R users who are aware of the existence of the many useful functions in an ever growing number of packages. The fact that R modules are findable, allows a broad community of internet users to use and cite them. Citations of R modules in research papers can (among other measures) be used to measure academic impact. The number of citations that have been found so far (Borghers 2005 and http://www.wessa.net/wessacitations.wasp) clearly suggests that the cost per citation in the R framework is only a very small fraction of the cost that was computed in the hypothetical (but conservative) example in the previous section.

All R modules allow the user to change the underlying R code on the fly. This is not only important for seasoned R user-developers - it also implies that users without prior knowledge of the R language can now engage in experimentation and learning. It is too soon to draw any final conclusions but the feedback that we received from students so far, suggests that this approach helps to make the R language more accessible (and easier to learn).

The R framework plays an important role in the way it deals with session management. User sessions are stored on the server and contain a lot of information about user preferences and previous computations. The most visible feature that is related to session management is the history list. The user can retrieve previous computations (in various formats) without the need to have anything recomputed. The output results, statistical datasets, and all types of meta information of the R module are stored in the history list until the session expires (usually a few hours after inactivity). In addition, it is possible to permanently store computational results together with the meta data that was used to create the R module. As will be explained below, this effectively allows us to create archives of computations that can be reproduced and reused at any time.

## Infrastructure

The hardware infrastructure and the costs of maintenance are important aspects of the Business Model that are involved in operating a world wide system for statistical computation. In addition, the infrastructure needs to be scalable, secure, and robust. Scalability refers to the ability of the system to serve requests even if demand (the number of users and/or requests) grows. Security relates to the fact that the system administrators should be able to protect the servers from being flooded with meaningless requests – servers should only perform the tasks that were intended by the designer of the system. Robustness is the property of the system to function well even if there is a partial failure of the network of servers.

The R framework addresses these desirable features through the fact that the actual statistical computations are performed on a distributed network of dedicated R servers. At the same time costs related to infrastructure are moderate and can be shared among many entities. To better understand this issue, we have a look at what actually happens when a user finds an R module and submits a request.

Typically, a user pastes one or several data series of interest into the designated fields of the R module's HTML page which was accessed through an ordinary web browser. The user specifies the values of the parameters that have been included by the creator of the module, and submits the computational request. The main web server receives the HTTP-POST request and forwards it to the appropriate R module while the connection with the client remains open. The R module now generates preprocessed R code by inserting and formatting the data series and parameters. Also, a series of meta parameters is generated in order to identify the request and the output that is expected. When the R module finishes its tasks it stores the preprocessed R code in the web server cache and calls the (load-balancing) "director" software. The director maintains a list of distributed R servers with various types of usage and performance-related information. Based on the IP address of the user, past performance of R servers and other information it predicts which R server is the best candidate to perform the number crunching. The director sends a request to the selected R server and provides it with a unique URL that allows the R server to fetch the preprocessed R code through HTTP-GET. This call-back mechanism ensures that R servers only execute requests that have passed the security rules of the director. It also allows any institution that kindly provides an R server to effectively protect it with standard techniques.

Every R server contains a small PHP script that simply fetches the R code at the main web server and forwards it to the R engine. Maintenance of the R server software can be easily performed through a secure shell and only involves updates of the operating system or the R engine. A few experiments with old servers and virtual private servers have shown that this system works and performs reasonably well. The main point that makes this work, is the fact that CPU load is directly dependent on the director's assessment of the R server's ability to execute the request. When a large university wants to use this system it will not object to make a small contribution by having one or two R servers made available. The average cost (of infrastructure) per computation goes down with the number of institutions that are sharing each others resources.

When the R engine completes the computation it stores the output (HTML tables and pictures in postscript and png) in a local cache. The director software now fetches the HTML output and inserts it into an HTML template. The resulting page is sent back to the user and the connection is closed. The browser of the end-user interprets the HTML code and discovers one or several pictures that are fetched from the R server's cache. The user is able to view the raw input and output files of the R engine that performed the computation through a link in the result page that points to the R code in the R server's cache.

**Financial aspects**

The costs that are related to the Product and Infrastructure are low and shareable thanks to the Framework's design. The (non-)financial benefits for Customers and Producers are clear and measurable under the conditions that have been outlined. However, a few remaining issues still need to be addressed to make the picture about the financial aspects of the Business Model complete.

It is clear that any open-access project like the one that is described in this paper needs adequate funding on the long run - even if costs can be substantially reduced and the benefits are obvious. This problem however, is not endemic to open-access projects in academic communities. Many software projects in commercial companies fail or are abandoned due to lack of support, poor planning, insufficient funding, or because of less-than-expected sales figures. When a Business Model is adopted this does in no way imply that Management should not play an active or even an anticipating role that makes sure the project is able to evolve and adapt to ever changing circumstances. At this moment, it is uncertain what types of revenue streams will be employed to make this project sustainable. However, it is very likely that a combination of revenues will be necessary to continue the provision of free and open access computing services to the world. Let me briefly list a few ideas that are currently under consideration:

- funding for educational projects based on the Framework
- funding for related research projects
- small donations from institutions in exchange for additional benefits
- moderate and relevant advertising through third-party providers
- hardware sponsoring by companies from the ICT and Consulting sectors
- consulting fees for the creation of tailor-made solutions for commercial (non-academic) use
- usage or time-based fees for commercial (non-academic) companies

Each of these ideas have been tested in the past and promise to be effective. In any case, the creation of a non-profit foundation will be required to manage these revenues and make important decisions about future directions.

The first idea in the above list deserves some additional comments. Thanks to the past successes of the Xycoon Project (in terms of web traffic and citations) we were able to obtain funding for the development of a new educational software project that consists of a Compendium Platform based on the described R Framework. This is a nice illustration of how the development of a statistical computing framework can be rewarded by a funding agency because of the fact that the output is directly usable in education. More importantly, the institutions that are involved in this particular project (they all belong to the Catholic University of Leuven Association) have made a binding commitment to support the continuation of the new project on the long run in terms of infrastructure and maintenance. Finally, it is important to emphasize that the proposed R Framework is not limited to providing only a user interface for the R language – for this purpose it would make much more sense to use one of several excellent systems that are currently available for the creation of sophisticated GUIs. Rather, the Framework should be seen as an attempt to bring some business sense to open-access computing, which promises to effectively disseminate scientific results for educational and research purposes at a low and shareable cost.

# Reproducible and Reusable Research

Future work will focus on the creation of a platform for compendium publishing that is based on archived computations, generated by R modules. Instead of building compendia based on articles, with extensions containing the statistical data and software to reproduce the results, we envision a system where the computational output is the core object. The main point in making the computation the core object is that it allows any author to use the archive, even if the article is published in a closed-access journal. If compendia would be based on articles (as core object) then the underlying technology would be disruptive for the traditional business model of academic publishing. In my view it is better to avoid such a clash and to use the computation-based compendium as a means to make research reproducible and reusable, regardless of the underlying publishing model.

Any computation that is produced by an R module can be easily stored in a time-stamped archive. Everything else that is needed to reproduce the result is contained in the meta data: the underlying R code, data series, parameters, etc... Articles that rely on the computation may be added to the meta database – even comments and discussions between students or researchers can be part of the compendium. Any article can make reference to the archived computation by the use of a permanent URL that can be used by the reader to view the original output. In addition, the reader will be able to exactly reproduce the computation or to have everything recomputed with different parameters, datasets, or R code. This effectively allows any knowledgeable user to reuse the research results and create new "derived" R modules that would have the status of a "child" of the original computation. In effect, this system does not only make research reusable, it also allows anyone to track progress of on-going research through a hierarchical tree of parent-child relationships. The consequences for science dissemination and academic education can be quite substantial.

As an example, let us briefly think about how hard it is to create a so-called "constructivist" learning environment (in a student-centered approach to statistics education) when the number of students is large. In the context of the social constructivist paradigm, student learning occurs mainly through interaction, discussion, group work, etc... Students with a "connected" attitude towards learning are believed to benefit from such a learning environment. One technique that allows us to introduce social construction of statistical knowledge is Peer Assessment. Let us suppose that the educator instructs students to do assignments with a weekly time interval. After submission of the assignment, the educator explains about commonly made mistakes and discusses one or several solutions. In the next week, students are required to work on the next assignment and at the same time they perform some form of Peer Assessment. This process generates a lot of interaction and discussions where students can learn from each other. The problem with these kinds of group activities is that the work of peers is very hard (or at best extremely time consuming) to reproduce. The R Framework can be used to create a Platform that archives each computation of an arbitrary number of students in the form of meta data objects. These objects can be viewed by other students and by the educator – and at any time it is possible to have the R Framework reproduce the result that was described in the paper that is under review. More importantly, it is possible to make minor (or even large) changes to the data, the parameters, or the underlying R code before the analysis is recomputed. This mechanism effectively allows students to reproduce, reuse and review work from any computation that was produced with R modules. In addition, the educator has the means to investigate the actual learning process, whereas normally she would only observe the output that was submitted.

As explained before, this new project received funding because it allows us to build useful applications for education. However, it takes only little imagination to see many possible applications in the domain of collaborate scientific research or even publishing. At this point we can only guess what types of new opportunities will emerge when we empower authors to easily and freely create R modules that make their research reproducible and reusable. If the R Framework works for education (as described above) it automatically opens opportunities in research. Therefore, the Business Model of the proposed Framework primarily targets educational applications even if students do not pay to use the system. It is time to engage in a friendly form of competition with the large commercial software producers that build long term relationships through push-pull Marketing strategies. Friendly competition ultimately leads to better dissemination of research, more interaction in statistics education, and the adoption of new methodology in commercially produced statistics software on the long run.

## Acknowledgements

I would like to express my gratitude to Ed van Stee, and both referees who have contributed considerably with challenging questions and very useful suggestions.

## References

Constantinides Efthymios (2002), The 4S Web-Marketing Mix model, *Electronic Commerce Research and Applications* (1), 57-76

Borghers, E, and P. Wessa (2005), The Xycoon Project, Research Paper 2005023, Faculty of Applied Economics, University of Antwerp, 41 pages

Edwards K. (2005), An economic perspective on software licenses - open source, maintainers and user-developers, *Telematics and Informatics* 22, 111-133

Johnson, J.P. (2001), Economics of Open Source Software, [Website] http://opensource.mit.edu/papers/johnsonopensource.pdf, downloaded 2007/01/23

Lucy, D. Aykroyd, R.G. & Pollard, A.M. (2002), Non-parametric calibration for age estimation. *Applied Statistics* 51(2): 183-196

Osterwalder, Alexander (2004), The Business Model Ontology: A Proposition in a Design Science Approach, PhD dissertation, Ecole des Hautes Etudes Commerciales de l'Université de Lausanne, Switzerland

R Development Core Team (2006), R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/

Rubinstein, H. (2002), All About Electronic Scientific Publication, Nobel Symposium (NS 120), in *Virtual Museums and Public Understanding of Science and Culture*, May 26-29, 2002, Stockholm, Sweden

Thomson Scientific - ISI Essential Science Indicators, "Average Citation Rates for papers published by field, 1994-2004, in Mathematics", [Website] http://in-cites.com/analysis/04-fifth-math.html#Highly%20Cited%20Papers, downloaded 2005/11/25 - 09:41 PM

Välimäki, Mikko, Oksanen Ville (2005), The impact of free and open source licensing on operating system software markets, *Telematics and Informatics* 22, 97-110

Wessa, (2007), Bivariate Kernel Density Estimation (v1.0.5) in Free Statistics

Software (v1.1.22-r1), Office for Research Development and Education, URL http://www.wessa.net/rwasp_bidensity.wasp/