Open Source Software Possibilities for Caribbean Researchers

By

Christopher Martin Clarke

19 November 2001

Open Source Software Possibilities for Caribbean Researchers

Introduction

The small research institutions have typically not paid sufficient attention to the true cost of the software that they use. For one thing, small size and the relative lack of sophistication of these institutions often meant that there was a preference for generic desktop type/single user systems such as office suites, single user econometrics packages, and desktop DBMS. In most cases predatory pricing policies of the monopoly provider (Microsoft), the relatively liberal interpretation of the licensing provisions by Caribbean research institutions and the lack of adequate provisioning for training and development meant that software represented a relatively small part of the budget for computerization in the organization. The striking thing about many of the early computerization plans was that most of the emphasis was given to infrastructure (cabling, ducting, desks, shelving etc.) and hardware (workstations and servers) while the issues development and training only received cursory attention.

Attempts to introduce multi-user software systems (Oracle, Fame, Aremos or SAS.) where a recurrent annual license is required were often met with greater or lesser degrees of resistance. It was often necessary to show how such an investment was often could be aromatized over a number of different cost centers in the organization. Alternatively, some organizations were able to enter into special arrangements with International institutions for example UNCTAD – DEMFAS debt monitoring software or IMF – Aremos that significantly reduced the cost of the software. Unfortunately these arrangements have not been very stable.

The last three years or four years have seen dramatic change in this situation as the main player in the desktop productivity market has adopted more aggressive pricing and licensing policies. Concerns have also been raised about the reliability and security of proprietary software these packages (IIS, Ms Outlook etc.) have been shown to be extremely venerable to attack by malicious programmers. As a consequence users are being forced to revaluate the their current assumptions IT strategy. This partially explains the growing interest of what was initially called the "free software" movement but what has become to be known called open source software. Open source packages like Linux, Free BSD, My SQL and Apache have been adopted by a broad array of organizations ranging from small cash strapped business and non-profits to giant corporate entities like IBM, Oracle and even Microsoft. In this paper we make the case not just for more use of open source application in Caribbean research institutions but also for the use of open source development methodologies in large collaborative projects. The paper is divided into five (5) sections. In the next section we examine some disturbing trends in the proprietary software industry that have negatively impacted customers. This is followed a brief examination of the economic and commercial basis for open source. Next we present a short primer on open source movement addressing issues such as licensing, distribution, support and so on. In section four we look at some of the open source applications that should be of interest to researchers in economics and finance. Finally, we present some suggestions for deciding when to deploy open source applications i

Problems with Proprietary Desktop Software

Microsoft currently dominates the desktop productivity market. Having consolidated its desktop monopoly in the early part to the 1990s Microsoft has been gradually increasing prices and moving away from the perpetual license to a subscription model which is norm for in multi user software. At the same time Microsoft is also being much more aggressive about enforcing its proprietary rights. In some of the islands (Trinidad, Barbados Jamaica) it has set up organization called Business Software Alliance dedicated to the stamping out of "software piracy". This organization have been targeting businesses (comprehensive software audit) to ensure that license fees are collected.

In contrast, while software costs have risen dramatically, the costs of hardware and infrastructure have fallen dramatically. On the desktop side the situation could be best illustrated by the fact that that in 1997, the operating system/office suite represented 20% of the cost of the typical business PC while in 2001 this represented 50 % of the direct cost of the standard desktop PC[1]. These calculations of course ignore other software related costs such as training, maintenance and development efforts over the lifetime of the PC (usually called the Total Cost of Ownership – TCO). In the networking field developments such a wireless LAN, the pervasiveness of the Internet and the growth broadband have greatly served to reduce the cost of infrastructure.

Another issue facing the users of proprietary software is that licensing regime is becoming increasingly restrictive. It is well known that the Microsoft EULA severely limits the users right to transfer/sell or even give away installed software. Recent developments include; the elimination of the version upgrades (usually the cheapest upgrade mechanism) the attempt to encourage force users to adopt the software release that Microsoft deems "current" even if they are not interested in the any of the new features of current release. Furthermore, in the case of the Windows XP release, Microsoft is also attempting to tie the software to specific hardware configuration. This feature is known as "activation". Users interested in "activating" their software must first contact Microsoft. Any significant change in the PC's hardware profile also requires re-activation and a call to Microsoft. Aside form drastically restricting the freedom of users and increasing the administrative burden, these developments also have significant costs implications. Indeed, the Gartner group estimates that recent changes in Microsoft's licensing strategy increase in licensing cost for corporate users of ranging from 35% to 109% depending on the frequency of the upgrade cycle.

At the same time the license regime has becoming more restrictive and expensive, users have found that the quantity and quality of support and documentation available as part of the initial software purchase has diminished significantly. The user is either forced to pay for any "extra" direct support as add on or to rely on ad-hoc Internet postings for for support.

[1] Own calculation based prices from various issues of Computer Shopper magazine

Many of the larger users have become particularly concerned about the vulnerability of proprietary software to malicious attacks. Indeed John Pescatore of the Gartner Group, writing in the wake of a continuous series of attacks on Microsoft's Internet Information Server (IIS) and other related software recently recommended that businesses should start actively investigating and deploying alternatives to such software. Suggested alternatives include such as the openings from Iplanet and the open-source Apache[2].

At the macro level one also has to address issues of technology transfer and the role of intellectual property rights the development process. These issues were quite popular among Caribbean economist in the 1980s however, in recent times they seem to have fallen out of fashion. Economist like Pantin and Farrell examined the impact of the regime of patents and technology licensing on issues such as foreign exchange flows, private sector debt and the negative impact on indigenous technological innovation. We believe that while the terminology may be outdated the arguments raised by these authors are still valid. Anecdotal evidence from a wide array of sectors tends to support the contention that so called "computerization" based on the wide-spread use of proprietary desktop software where there under investment in indigenous software development and training has failed deliver any significant long term gains in organizational productivity. In this context, Caribbean policy makers should probably pay more attention to the "learning" effects gained from more local involvement in the software development process and less attention upholding intellectual property rights of large monopolist

[2] Gartner Viewpoint September 21, 2001

The case for open-source

In the last section we examined some of the arguments against proprietary software. In the past the consumer could do very little about this situation. The alternatives were to find another vendor or to build your own bespoke software. The former was increasing hard to find as Microsoft decimated the market and in any case there was no guarantee that the new vendor would not behave in a similar fashion to Microsoft once the consumer had become lock into the product. In the latter case this strategy was usually beyond the means of small players and also left the customer dependent on a particular internal or external developer. More recently, the wide spread acceptance of products like Linux and Apache have herald the arrival of open-source or as it is sometimes known as the "free" software movement as a viable alternative to proprietary software., In contrast to proprietary alternatives open-source software usually comes with a license that allows the user to make as many copies as she wants. She can even give it away or even sell it to another user without having to obtain the author's permission. Moreover, the user also gets access to the source code and in most cases the monetary cost of the package is zero. Even more astonishing is the fact that open-source software has been shown to be more reliable than its proprietary competition even though it is produced by groups volunteer programmers scattered all over the world linked together by the internet. To the non-technical user this sounds too good to be true and it is important for the advocates of open-source to present a coherent case for the economic rationality of open-source as well as some commercial arguments in favour of it. This is the purpose of this section.

The Economic Case

In a neoclassical economy, goods are priced equal to their marginal costs to produce. But there is a serious flaw in this mechanism when applied to information. The software industry is characterized by high fixed cost but the marginal cost of duplicating software is very small, if it is priced equal to its marginal cost, the price p will be near zero in competitive equilibrium. Average cost will be nearly equal to average fixed cost which will usually be well above the equilibrium price in the long run. In such circumstances, the profits will be negative and nobody will produce information in equilibrium. This is the famous paradox presented by Arrow (1962), but he didn't think it so fatal. From a theoretical perspective, it can be decomposed to two problems - ex ante adequate pricing to recoup fixed costs and ex post efficient distribution - and solved separately. For example, if the government buy inventions with optimal prices and distribute them freely, the first best can be attained. A there is no ideal "auctioneer" to price information optimally without markets, the social cost of direct intervention is much bigger than its benefit. The most popular policy is legal protection of the "intellectual property right" (IPR) to create innovator's monopoly. But monopoly distorts resource allocation, so the net social gain from IPR depends on whether the appropriation effect in which imitation reduces innovator's ex ante incentives, is larger than the learning effect, ex post efficiency resulting from information sharing.

Learning effects are even larger in software, because, in writing programs, all programmers learn from other source codes and sometimes copy them in part. They can learn nothing from binaries. If they were all forced to write software from scratch, the resulting efficiency lopen-source from duplication would overwhelm the gains from protecting IPR.

Moreover, Raymond (1998) argues that a programmer's incentive to innovate is an increasing function of the number of copies of her works in the active communities of open-source, such as and Apache, the most popular WWW server software. Programmers "homestead" cyberspace to write programs, for which they claim "ownership" of originality. You can see long credits of programmers at the end of release notes of open-source. This kind of joint ownership is inefficient for physical assets, but efficient for non-rival assets such as information. Taking into account network externalities, the net benefit of protecting IPR is dubious for software. If learning effects are greater than appropriation effects, it might be socially efficient not to protect IPR for software, as insisted by open-source proponents such as the GNU project.

The Commercial Basis for open-source

The case for open-source can also been made on pure commercial grounds. One of the main benefits of open source software is that it tends to be more reliable than its commercial alternatives. This is largely a function of the fact that mature open-source software only wins aceptence if has been subject to rigorous peer review. This also means increased security; because code is in the public view it will be exposed to extreme scrutiny, with problems being found and fixed instead of being kept secret until the wrong person discovers them

The Open Source Institute has also argued open-source shifts the balance of power away from the vendor and back the customer. Open source empowers the customers by reducing their dependence on a single vendor. Because the customer has access to the to the source, he can better survive the collapse of the vendor. The customer is no longer totally at the mercy of unfixed bugs and is not shackled to every strategic decision the vendor makes. Moreover, if the vendor's support fees become exorbitant, the customer can buy support from elsewhere. In a similar manner customers can also benefit by opening the source of internal projects that are meant for internal use and for sale on the market. This could speed up the development by enlarging the pool of human resources available to the project and more importantly help to insure that there is always a cadre of programmers both inside and outside the company familiar with the software.

Since open all open source software can be freely copied the adoption of open source reduces the administrative burden associated with the licensing regime. Using most commercial software involves software licenses, and tracking software copies and usage. This demands record keeping, and legal exposure. Both raise costs. Thus, juggling software licenses and copies is a source of costs to businesses, and legal risk to businesses and individuals. In most cases, such tracking is imperfect, sometimes intentionally, usually not. Any such imperfection exposes the guilty party to legal actions (fines, litigation, arrest) due to breaking laws and violating copyrights; an intellectual property quagmire. Most/all open source software can be freely copied and used. There are no licenses to track and thus no related costs, or legal risks.

Open Source Primer

The open-source sector differs radically from the proprietary software industry. It traditions of openness, standards, use of the more obscure internet tools such as Usenet, mailing list archives, makefiles, Unix and so on can be quite alien to PC users weaned on Microsoft's vision of computing., In this section we present a brief examination of the mechanics of open-source "industry". We start by defining what open-source is. Open source software is software that comes with permission for anyone to use, copy, and distribute, either verbatim or with modifications, either gratis or for a fee. In this case "free" refers to be to the use to which the software can be put to uses as opposed to the price one pays for of the software. This is summarized in the mantra

"Free as in free speech as opposed to free as in free beer"

While in practice there may be differences in the specific license terms under which open-source can be distributed, the common factor is that source code must be made available to the user. Indeed, another attraction of the open source movement is the collaborative nature of the development process. In successful open-source project, a community of developers from all over the world must work together to build the software. This promotes software reliability and quality by supporting independent peer review and rapid evolution of source code. The knowledge is shared among this community and this serves to ensure that the customer is not dependent on any particular developer or group of developers.

Open source software has a undeserved reputation as being "geek-ware" as the focus of the early efforts was to build the basic tool such as compilers (GCC) and the operating systems then GNU Linux and both of which has gained wide spread acceptance in the market place. Indeed open source is underlies the basic infrastructure of the Internet where it has been used to develop technologies such as DNS, Sendmail, various TCP/IP stacks and utility suites, and scripting languages such as Pearl and Python.

However, open-source is not just limited to compiling tools, operating systems and Internet protocols it has representatives in every category of software. The Netcraft web server which tallies web servers in use on the Internet consistently shows that the open-source Apache web server to have over 50% of this market. In the multi user database category MySQL enjoys wide popularity for e-commerce applications. Additionally, versions of the leading commercial databases such as Oracle, Sybase and DB2 all currently run on the Linux platform.

Some Terminology

One source of confusion for the new comer to open-source software is that while the principle of "free speech not free beer" is a fairly straight forward, in practice there are in fact a number of different issues of detail that have to be addressed to operationalize this concept. These include issues such as whether derivative works are also "free", if the product could be distributed commercially, whether restrictions can be placed on who can modify the code and so on. This has lead to a proliferation of many different kinds of open-source licenses on the market. More recently, attempts to commercialize open source have lead to the term "open-source" been used more as a marketing slogan. A number of companies have been floated on the stock markets to sell open-source solutions while established software vendors have adopted open source as a development strategy. This has resulted in a situation where the terms "free" and open-source being used when what is being offered is not actually be "free". This section attempts to clarify some of these issues

Diagram below taken by Chao-Kuei presents the different kinds of software that may be typically available for free download on the Internet.

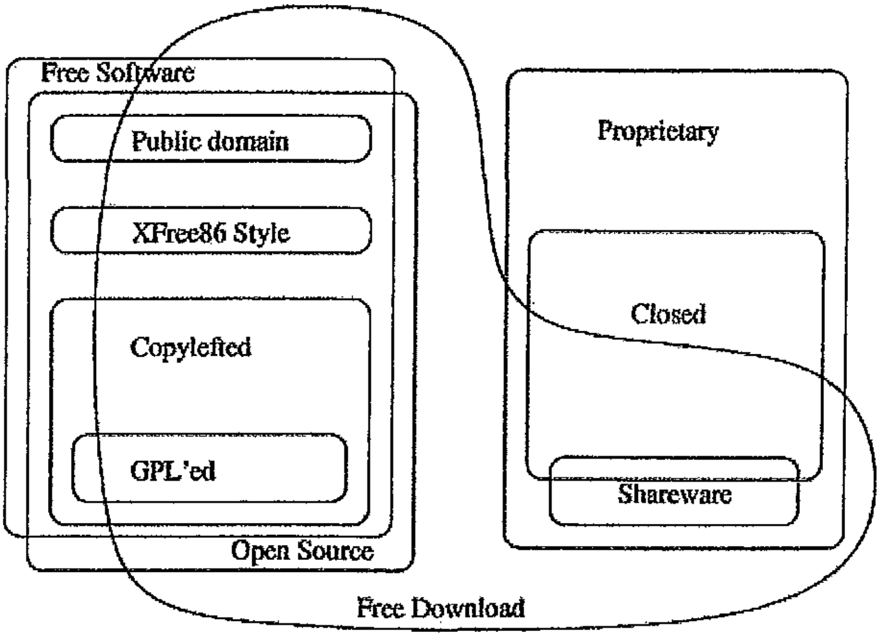

Figure 1 Categories of Software available for free download

Free software vs Open Source Software

We saw in our discussion above that the essential characteristics of open source software are the availability of the source code and the fact that there should be no restrictions on the use to which the software could be put. In practice there are many different ways to make a program free many questions of detail, which could be decided in more than one way and still make the program free. The "free software" movement is attributable to Richard Stallman a computer programmer with strong ideological and ethical objections to the notion proprietary software. Stallman set up the Free Software Foundation to promote the development of "free software". The Free Software Foundation (FSF) is associated with the GNU project – GNU's not Unix and has pioneered the notion of "copyleft" which is discussed in more detail below.

More recently, in 1998, the open-source movement was formed partially in response to the decision of Netscape to release the source code for its browser called Mozilla.. In contrast to the FSF, the Open Source Institute makes the case for open-source on the basis of the commercial benefits that flow to developers, customers and software companies. In practical terms there is very little difference between the "GNU Licensing" and the those proposed by OSI

Public domain software

Public domain software is software that is not copyrighted. Some copies or modified versions may not be free at all. Public domain software is a legal term and means, precisely, "not copyrighted".

Copylefted software

Copylefted software is free software whose distribution terms do not let redistributors add any additional restrictions when they redistribute or modify the software. This means that every copy of the software, even if it has been modified, must be free software. The GNU Public license (GPL) from the Free Software foundations is the most popular way to implement copyleft. This license ensures that any derivative work must also be GPL. This aspect of license has been attacked by Microsoft founder Bill Gates as being "viral" in the sense linking of GPL code to Microsoft APIs supposedly attacks Microsoft copyright and even the basis of the free enterprise system. In practice this is a bit of a red herring, as copyleft also supports the notion of the Lesser General Public License LGPL to deal with the need to link to code that is distributed under more restrictive conditions than the GPL. Code licensed under the LGPL can be dynamically or statically linked to any other code, regardless of its license, as long as users are allowed to run debuggers on the combined program. In effect, this license recognizes kind of a boundary between the LGPL'd code and the code that is linked to it.

Non Copyleft "free" Software

The GPL and LGPL are generally regarded as being close to the ideal methods for open source licensing. However, other kinds of "free" license also schemes also exists. Many of these predated the GPL and in many cases these are considered to be compatible with it. Some popular free licenses compatible with copyleft include; the modified FreeBSD license (used for the popular FreeBSD operating system), The X11 license (a powerful graphics library used in the software used to run most graphics cards) and the current licenses for the languages Python and Pearl.

Additionally, there a number of other "free" licenses, which while not completely compatible with copyleft but are still considered to be "free". Theses include; the IBM Public License, Version 1.0 – Used by IBM for its open source projects, the LaTeX Project Public License which governs the terms under which software based on the TeX typesetting system invented by D.E. Knuth is distributed, the various licenses for the Apache web server license and the Mozilla Public Licence which governs the opens-source version of the Netscape client. In most cases these incompatibilities result from purely technical reasons of detail such as the requirement to display copyright notices and so on. However, in some cases such as the Netscape Public License the license gives the original author the right to incorporate the modifications from other authors in proprietary versions of the software.

In contrast, there are a number of licensing schemes that claim to be free or public or open but in fact do not grant the user unrestricted rights to the software. A number of these schemes have been set up by commercial software vendors hoping to cash in on the free software movement. These include schemes such as the Apple Public Source License (APSL) and the Sun Solaris Source Code (Foundation Release) License. Both these licenses place severe restriction on the commercial use of the software. By contrast the GPL license was developed to ensure that no barriers are placed on the commercial exploitation of "free" software except that any derivative products must also be free.

Semi-free software

Semi-free software is software that is not free, but comes with permission for individuals to use, copy, distribute, and modify (including distribution of modified versions) for non-profit purposes. For example many products are distributed under something called a

Community license. The terms and conditions of the various communities differ but the Sun Community source license is a typical. In this model the users "share" what is for all intents and purposes a modified proprietary license. This license tries to distinguish between different kinds of users such as students, researchers, internal developers, commercial developers and so on, places restriction

Proprietary software

Proprietary software is software that is not free or semi-free. Its use, redistribution or modification is prohibited, or requires you to ask for permission, or is restricted so much that you effectively can't do it freely.

Freeware

The term "freeware" has no clear accepted definition, but it is commonly used for packages which permit redistribution but not modification (and their source code is not available). These packages are not free software, so that "freeware" is not free software software.

Shareware

Shareware is software that comes with permission for people to redistribute copies, but says that anyone who continues to use a copy is required to pay a license fee. Shareware is not free software, or even semi-free. For most shareware, the source code is not available; thus, you cannot modify the program at all. Shareware does not come with permission to make a copy and install it without paying a license fee, not even for individuals engaging in nonprofit activity. (In practice, people often disregard the distribution terms and do this anyway, but the terms don't permit it.)

Commercial Software

Commercial software is software being developed by a business, which aims to make money from the use of the software. "Commercial" and "Proprietary" and not the same thing. Most commercial software is proprietary, but there is commercial free software, and there is non-commercial non-free software. For example, many Linux distributions are such a RedHat and Mandrake under the terms of the GNU GPL, and every copy is free software; but its developers sell support contracts, documentation and so on.

Organization of Open Source Projects

Open source development is very decentralized process relying on informal groups of programmers. These volunteer groups are loosely tied virtual communities, where membership changes are very frequently. This is in contrast to usual forms of organization such as firms where the boundary is crucial for their governance, but this *openness* is the core of the mechanism to govern open-source groups. The organization, overall design and direction of any project is usually in the hands of one or two programmers who act as "benevolent dictators" with the final say over what is available in the released version of the software. If a project can't attract enough attention, its participants will exit and the project will come to a halt. The effectiveness of an open-source project doesn't depend on any democratic procedures such as majority voting but "natural selection": anybody can join it, but few can survive the hard process of never-ending discussions and co-development.

Obtaining Open Source Software

Most open-source software can be obtained directly by download from the Internet. A system of mirror sites in the major countries serves to ensure that download times are not prohibitive. There are currently no Caribbean mirror sites. Users generally start by visiting the website of the open source/free software organizations such as the GNU web (www.gnu.org) or a specialist site like www.mathtools.org. These sites usually provide categorized lists of the various projects. The user then link to an individual projects site. Users must be aware that the open-source movement is quite fragmented so the lists provided may not be comprehensive. Users should use search engines such as www.google.com to supplement the portal sites.

The open-source project's website will usually contain links to information about the goals, aims and philosophy underlying, project binaries targeting the various platforms (Windows, MacOs, Linux etc.), documentation including electronic version of manuals, "bug lists" and general support information such as FAQs, change logs and mailing lists. Additionally, a source code repository is available via the Concurrent Versions System CVS.

Binaries and source code are also available in CD form directly from the authors of software such as Redhat, Mandrake and Zope. These vendors (for example Redhat, Zope and Mandrake) also provide email support and printed documentation retail users for a limited time period. There are also a number of commercial redistributors whose role is simply to provide CD images of the source code and binaries downloaded from the project's website. The user pays a nominal fee to cover the duplication and shipping. This is often a best option when Internet connections are slow. Some of the more commercially oriented projects such as RedHat or Cygnus also provide value added services such as proprietary add-ons, more extensive support options, consulting training and certification.

Support and Documentation

As discussed above a few commercial open-source vendors provide "traditional" support (email support, printed manuals etc.) on the purchase of the retail CD versions of the software. However, most open-source users rely heavily on decentralized Internet based support. This is provided both by the authors in the form of electronic manuals, mail list archives and FAQs and by informal groups of users via Usenet and the Web. Additionally, some commercial publishers such as O'Reilly Associates have specialized in catering to the open source market. O'Reilly commissions programmers and others engaged in open-source projects to create user guides and other manuals for the most popular open source projects. Indeed, it can be argued that aside from some niche products with well defined (mall) markets there has been a convergencein the support options available for proprietary and open-source users.

Some Open Source Tools for Economic Researchers

In this section we briefly examine some of the open source tools that may be of particular interest to economists. The list of products discussed is this section is by no means comprehensive and the reader is welcome to provide addition suggestions.

LaTeX

LaTeX is the open source implementation of TeX typesetting language developed by Donald E. Knutt. Because of its facility with mathematical formulae LaTeX has become the standard for submitting articles to academic journals in the "hard" sciences. LaTeX has been described as a poor mans desktop publishing system. However, the LaTeX provides much more than this. In a LaTeX environment, LaTeX takes the role of the book designer and uses TeX as its typesetter. But LaTeX is "only" a program and therefore needs more guidance. The author has to provide additional information which describes the logical structure of his work. This information is written into the text as "LaTeX commands.".

This is quite different from the WYSIWYG approach which most modern word processors such as MS Word or Corel WordPerfect take. With these applications, authors specify the document layout interactively while typing text into the computer. All along the way, they can see on the screen how the final work will look when it is printed.

The main advantages of LaTeX include

- Professionally crafted layouts are available, which make a document really look as if "printed."
- The typesetting of mathematical formulae is supported in a convenient way.
- The user only needs to learn a few easy-to-understand commands which specify the logical structure of a document. They almost never need to tinker with the actual layout of the document.
- Even complex structures such as footnotes, references, table of contents, and bibliographies can be generated easily.

- Free add-on packages exist for many typographical tasks not directly supported by basic LaTeX. For example, packages are available to include PostScript graphics or to typeset bibliographies conforming to exact standards.
- LaTeX encourages authors to write well-structured texts, because this is how LaTeX works—by specifying structure.
- TEX, the formatting engine of LATEX2", is highly portable and free. Therefore the system runs on almost any hardware platform available.

Gretl

Is a software package for econometric analysis, written in the C programming language. It is distributed under General Public License (GPL) Foundation. Gretl is a fairly new package that comprises a shared library, a command-line client program, and a graphical client built using GTK+.

The features in the latest release include:

- A reasonably wide variety of least-squares based estimators (including two-stage least squares).
- Easy intuitive interface.
- Single commands to launch things like augmented Dickey-Fuller test, Chow test for structural stability, Vector Autoregression.
- Reads own format ascii data files, Comma Separated Values files, BOX1 files, own format binary databases (allowing mixed data frequencies and series lengths) and also RATS 4 databases.
- Output models as LaTeX files, in tabular or equation format (not very flexible yet).
- Integrated scripting language: enter commands either via the gui or via script.
- Command loop structure for Monte Carlo simulations.
- GUI controller for fine-tuning Gnuplot graphs.
- Link to GNU R for further data analysis.

The website lists the following the following limitations for the package

- Not much in the way of maximum likelihood methods yet (only logit and probit). I'll work on this if there's enough interest in gretl.
- Not much in the way of nonlinear least squares either, though an iterative procedure is available for linearized models.

GNU R

GNU R can be regarded as an implementation of the S language that was originally developed at Bell Laboratories by Rick Becker, John Chambers and Allan Wilks, and also forms the basis of the S-Plus systems. S and its derivatives are extremely popular in the field of finance. In contrast to the Gretl project the R is quite mature project. Additionally, since it is an implementation of the S language programs written for S-Plus can be run with little modification. There is an important difference in philosophy between S (and hence R) and the other statistical systems such as SAS and SPSS. In S a statistical analysis is normally done as a series of steps, with intermediate results being stored in objects. Thus, whereas will give SPSS and SAS copious output from a regression or discriminant analysis, R will give minimal output and store the results in a t object for subsequent interrogation by further R functions. R provides the following features

- An effective data handling and storage facility
- A suite of operators for calculations on arrays, in particular matrices
- A large, coherent, integrated collection of intermediate tools for data analysis
- Gaphical facilities for data analysis and display either directly at the computer or on hardcopy
- A well developed, simple and effective programming language which includes conditionals, loops, user defined recursive functions and input and output facilities.

Open Source Office suites

One of the more exiting developments in the open-source movement has been the development of open-source alternatives to proprietary office suites. Currently there seem to be two main ongoing projects in this area

- The Open Office initiative www.openoffice.org
- Koffice - www.koffice.com

The Open Office project was founded when Sun decided to open-source its StarOffice code. StarOffice was essentially an attempt to create a clone of MS Office. This project has put a lot of emphasis on issues such as standardization focusing on questions such as the development of open file formats and support for industry standards such as XML. The latest version of the software suite includes the following primary applications

- StarOfficeWriter provides word processing capabilities including HTML authoring.
- StarOffice Calc provides comprehensive and highly programmable spreadsheet functionality with hundreds of built-in functions.
- StarOffice Impress is a full-featured presentation application with advanced drawing functionality that gives users greater control and flexibility when creating impactful presentations

KOffice is a free, integrated office suite for KDE, the K Desktop Environment a popular windowing environment on Linux.

The current KOffice 1.1 release features the following applications:

- KWord - A frame-based word processor capable of professional standard documents
- KSpread - A powerful spreadsheet application.
- KPresenter - A full-featured presentation program.
- Kivio - A Visio®-style flowcharting application.
- Kontour - A vector drawing application.
- Krayon - A pixel-based image manipulation program like The GIMP or Adobe© Photoshop®.

- Kugar - A tool for generating business quality reports.
- KChart - An integrated graph and chart drawing tool.

Summary and Conclusion

This paper examined the potential of open-source software in Caribbean research organizations. We saw that the growing interest of open-source among both large and small organizations is a consequence of the convergence of a number of factors including the attempts by some vendors to exploit monopoly positions, restrictive licensing practices, the need to reduce reliance on a single vendor or group of developers and growing concerns about the reliability and security of proprietary software. Indeed we· also economic arguments that tried to show that the software business is a bad fit for the notion of IPR. Indeed the application of these rights in the field of software development is inefficient as it reduces innovation and has negative effects on "learning" among software developers.

We then looked some of the workings of the open-source industry examining issues such as licensing, distribution and support options. Finally we looked at some open source software titles that may be of interest to researchers in economics.

Finally, we caution that open-source is no "silver-bullet" in the sense that the user passively installs some software it and then expects it to solve the problems that are unique to the business. The only way for business maximizes the benefits from investments software is for then to channel sufficient resources to the areas of development, customization and training.