

Härdle, Wolfgang Karl; Horowitz, Joel L.

Working Paper

Internet based econometric computing

SFB 373 Discussion Paper, No. 1998,37

Provided in Cooperation with:

Collaborative Research Center 373: Quantification and Simulation of Economic Processes,
Humboldt University Berlin

Suggested Citation: Härdle, Wolfgang Karl; Horowitz, Joel L. (1998) : Internet based econometric computing, SFB 373 Discussion Paper, No. 1998,37, Humboldt University of Berlin, Interdisciplinary Research Project 373: Quantification and Simulation of Economic Processes, Berlin, <https://nbn-resolving.de/urn:nbn:de:kobv:11-10056812>

This Version is available at:

<https://hdl.handle.net/10419/61291>

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.

INTERNET BASED ECONOMETRIC COMPUTING

Version : 4/17/98 9:19:33

**W. Härdle
Institut für Statistik und Ökonometrie
Wirtschaftswissenschaftliche Fakultät
Spandauer Straße 1
Humboldt-Universität zu Berlin
10178 Berlin
Germany**

**J. Horowitz
Department of Economics
University of Iowa
Iowa City, IA 52242
USA**

Abstract

Modern econometrics requires implementation of highly specialized software. In contrast to mathematical arguments used in implementing new econometric techniques the corresponding software algorithms require specific platforms. The specialization of hardware and software, in fact, seriously impedes the adoption of new methods in applied research. It complicates the proliferation of new techniques and makes it difficult to motivate students to use the methods and to help students to develop an intuitive understanding of the methods in applications. We discuss the potential for reducing these problems through Internet-based econometric computing and instruction. We refer to existing examples of net-based teaching and present concrete examples for interactive teaching of elementary econometrics and statistics.

1. Introduction

An important characteristic of modern econometrics is the rapid development of new, mathematically complex methods whose implementation requires highly specific software and, sometimes, hardware. Software is typically written in the developer's favorite proprietary language and is available to others, if at all, only if they own the proprietary system. For example a program written in GAUSS can be run only by a user who has the GAUSS system on his computer and better has the same type of computer too. We believe that this specialization of hardware and software seriously impedes the adoption of new methods in applied research. Moreover, it greatly complicates the tasks of teaching students to use the methods

and helping students to develop an intuitive understanding of the performance of the methods in applications. This paper discusses the potential for reducing these problems through Internet-based econometric computing and instruction. We describe an existing network architecture and give concrete examples for interactive teaching of an elementary statistics course and for the dissemination of semiparametric methods.

Section 2 provides a more detailed discussion of the heterogeneity of hardware and software that creates barriers between researchers, potential users of new methods and students. We point out, however, that specialization of hardware and software can have benefits because it permits the developer to take advantage of unique features of hardware or software systems that are especially suited to the problem being solved. The methods for internet based computing that we discuss preserve these benefits. Section 3 explains what internet-based computing is and how it differs from the existing technology of downloading software from worldwide web (WWW) sites. Section 4 describes an existing architecture that implements these ideas and is in use in undergraduate teaching at the Humboldt-Universität zu Berlin and several other universities. Section 5 presents concluding comments.

2. The problem of heterogeneity

Three hardware platforms are in widespread use for statistical computing and graphical data interaction: Macintosh, UNIX, Windows. The first has a simple graphically oriented user interface and allows highly interactive dialogues with data. UNIX is used for high-speed and distributed computing but is often less satisfactory in graphical interaction. Windows aims at facilitating both high-speed computing and graphics but is weaker at present than UNIX for Internet access. Distributed computing is simply not possible under Windows unless one uses certain add-ons.

Many software platforms for econometric computing exist but are unfortunately not easily interchangeable. The reasons for this include the history of software development, the targeted user groups, and the optimization of certain software for specific hardware configurations. The original version of GAUSS (<http://www.aptech.com>), for example, was optimized for INTEL chips and, therefore, could not be transferred to Macs or UNIX platforms. Now GAUSS is available on UNIX, but the UNIX version does not have a graphical device that would allow, for e.g. interactive changes in the layout of graphs. SPLUS (<http://www.mathsoft.com/Splus>) was developed for UNIX systems and was only later transferred to PCs. Consequently, the PC version is different from the UNIX version. EVIEWS was developed for DOS and is now available for Windows but not for UNIX or Macs. TSP is a DOS program and is not easily transferred to a Windows/NT platform. SPSS exists for Windows but has still a batch structure that makes many mouse clicks necessary in order to generate implicitly the batch commands. STATA (<http://www.stata.com>) , SAS (<http://www.sas.com/>) and SHAZAM (<http://shazam.econ.ubc.ca>) are unusual in that mutually compatible versions exist for all platforms. Besides the software that we mentioned here as examples, there are many other platforms, but they also share the property of heterogeneity.

Heterogeneity of software platforms creates no problems if there is no need to exchange programs. Exchange of graphs, document files, and ASCII-based data sets can be carried out by FTP, provided that the user has the appropriate graphics plug-in and document reader (e.g., Ghostscript or Acrobat). However, there is also a need for exchangeable computer programs for implementing advanced econometric methods, as these are becoming increasingly complex mathematically, and writing the necessary programs can be a difficult and time-consuming task.

Graduate-level instruction in econometrics provides one example of the usefulness of exchangeability. It is not unusual for a faculty member at one university to give a short course at another. In some cases, a faculty member at one university may use electronic communication to present a course at several geographically dispersed locations. An econometric estimator may require heavy computing that is available on the researcher's home machine. During the course, modifications of this estimator and different applications may be discussed, and these may require access to the software at multiple locations. Exchangeability of software is necessary to enable students at all locations to carry out computational and empirical exercises that the instructor has prepared at his own university.

Collaboration among researchers at different locations provides another example of the desirability of exchangeability. In this case, the goal is to enable each collaborator to carry out computations using the same software. Ideally such cooperation should be based on a pool of easily accesible software and computing

power for all parties. For effective progress on a project that involves heterogeneous hardware and software, it is desirable for partners to have the ability to contribute methods despite being at different locations and working with different computing environments. In addition it may simply be a problem for a researcher who is a visitor in another establishment to be able to continue using his own programs.

On the other hand, heterogeneity of software does have the important advantage of enabling a developer of new methods to choose the software system that is best suited to the problem under consideration. Therefore the problem of exchangeability should not be solved by standardizing econometric software but by making software from different sources accessible to diversely equipped users.

3. Internet based econometric methods

The phrase "internet based econometric methods" can refer to several different concepts. One concept is to maintain applications programs on a server for users who have access to the software systems required to run the programs. The programs are downloaded by FTP and executed on the user's computer. An example is the STATLIB server of SPLUS at Carnegie-Mellon University (<http://lib.stat.cmu.edu/S/>). Other examples include the ELSA archive at the University of California, Berkeley (<http://elsa.berkeley.edu/>) and the CodEc software archive (<http://netec.mcc.ac.uk/CodEc.html>). These sites and others like them permit a user to retrieve programs quickly. The user must, however, have access to the software system that executes the programs (e.g. SPLUS for programs written in that language, GAUSS, FORTRAN, etc.).

STATLIB and ELSA also illustrate two different archival policies that are worthy of note. STATLIB accepts externally written applications programs with minimal requirements for formatting, documentation, and testing, whereas ELSA has relatively stringent requirements. The developer of a program in SPLUS can easily submit it to STATLIB, but the cost to a user of learning to use a STATLIB program may be high. Indeed, there is no guarantee that the program even works. In contrast, submitting a program to ELSA is relatively costly because of the documentation and testing that are required, but it is relatively easy for a user to implement a program that has been downloaded from ELSA.

One outlet for net-based proliferation of econometric methods is the Common Gateway Interface (CGI). This user interface allows outside internet users to enter text into a CGI window that may then be interpreted as data or program lines. An archive of GAUSS programs for econometricians with a CGI interface may be found at (<http://eclab.econ.pdx.edu/gpe/>). The CGI communication technique allows the distant user to send certain commands and thereby to try methods developed by others. It is not possible, though, to construct commands to read own data from the user's disk or display results in an interactive graphic. Own data may be entered only by hand or by cut/paste. Afterwards the data strings are sent to a serving computer. There exist numerous CGI interfaced net calculators. An XLISP-STAT (http://www.cern.ch/WebMaker/examples/xlisp/www/cldoc_1.html) based calculator may be found at (<http://www.stat.ucla.edu/calculators>).

An alternative way to do net based econometric computing is via Java. The Java technology enables the programmer to produce code that is independent of the user's operating system and applications software. Typically, Java is used to support interactivity in browsers. A Java applet that has been loaded into a browser can perform operations such as interactive graphics, least-squares mean-regression, and nonparametric density estimation. Essentially, a Java applet functions as an application module that operates on all machines without requiring the user to have a specific operating system or specific applications software. Several user interface tools are available for this technology. For example, sliders, point clicks, spreadsheets, and interactive drawing are available. The current technical implementation of Java is based on SUN's Java Development Kit (JDK) 1.1 (<http://www.javasoft.com/>). For Java applets to run in common browsers it is necessary to have the corresponding Java Runtime Environment (JRE) 1.1 enabled.

An example of a Java interaction based on point clicking, is the Support Vector Machine (SVM) running in the Royal Holloway College in London (<http://svm.dcs.rhbnc.ac.uk/pagesnew/1D-Reg.shtml>). One uses the mouse to enter points on a set of coordinate axes that are displayed on the screen. The SVM computes a nonparametric mean-regression using orthogonal polynomials. This regression application can be used to illustrate the effects of outliers and other pathologies of data. It cannot, however, be used for real-data applications because there is no capability for loading a user's data into the applet. The reason is that the browser supported JRE is not able to perform local file input and output (I/O).

Another example for educational interaction is the effect of bin width on the appearance of a histogram. This example has been developed by Phillip Stark at the University of California, Berkeley, and is available with many others at SticiGui© Java Tools (<http://www.stat.berkeley.edu/users/stark/Java/index.htm>). The user points with the mouse to a slider and moves a bar that is linked to the histogram bin width. A graph displays the histogram that is produced with the chosen bin width. As with the SVM examples, these are useful for illustrating the properties of statistical procedures in an instructional setting but cannot be used for real-data applications, since the user may not use his own data files.

A limited capability for data entry is provided by the procedures that are available at the Webstat Project at the University of South Carolina (<http://www.stat.sc.edu/~west/webstat/>). The user may enter data by typing them into a spreadsheet or by copying and pasting them from a file or downloading them by FTP from a server. Data entry by reading a local file is not available because applets loaded through a browser do not allow local I/O operations. Basic statistical operations may be performed, including computing means and variances, linear regression, plotting data. Graphs may be displayed via a menu-controlled user interface. One drawback of this approach is that the data-entry procedure makes use of large data sets infeasible. Another is that no source code is available to the user. Therefore, the user cannot modify the procedures to support special needs or extend them to classes of problems that they currently do not accommodate.

The foregoing interactions with statistical methods are most useful for teaching introductory econometrics and statistics students. An example is given by the virtual statistics laboratory of Rice University (http://www.ruf.rice.edu/~lane/stat_sim/index.html). In this example, the student

generates a scatterplot and then draws a regression line by eye. The display shows the residual mean-squared error and other goodness-of-fit statistics for the line that has been drawn. The student can then use the mouse to draw a new line and, by repeating this process, find the line that minimizes the mean-squared error. Other Java supported teaching methods are available at this virtual laboratory for correlation, sampling distributions, and approximations to the normal distribution, among others.

The Java technology used by Webstat and the laboratory at Rice University is not ideally suited to carrying out platform-independent, complex computations with large data sets. Technically, pure applets are loaded into the browser like plug-ins. They run independently of the Internet once they are loaded using the browser-provided Java runtime environment (JRE) Certain operations (e.g. I/O file transfer) are not supported by browsers although they are permitted in the Java language. The Java applet runs in a browser via a JRE that uses certain “classes” provided by the browser. A class is like a keyhole that accepts only certain keys, in this case Java commands. For security reasons free browsers do not come with an I/O class. One may, however, write Java applets that use local file I/O operations. In consequence, these must run independently of a browser. For realistic internet based econometric analysis it is, therefore, useful to start applets independently of the Java classes provided by a browser. Simple econometric computations can be done entirely by such an applet. For example, computation of means and variances, t-tests, and smoothing can be entirely written in Java and run on the user’s machine with the user’s data. One may imagine a whole econometric package written entirely in Java.

The disadvantage of this approach is that handling large data sets and carrying out intensive computing operations slows the user’s machine. The price for platform independent computing is that calculations written in Java are slower than those written in generic languages such as C or FORTRAN. In general, intensive computing is best performed on the provider’s server, whereas graphics and editing are best performed on the user’s machine. In other words, the Java applet technology is most useful when there is a good balance between the tasks carried out on the provider’s server and those carried out on user’s machine. This raises the question of scaling computing loads between a user and a server, the latter usually being the econometric method provider with a fast computer.

4. Client Server Econometric Computing

The foregoing discussion shows that a well designed client-server architecture should combine accessibility of methods on the Internet with computing loads that are distributed in a way that assigns tasks to client and server machines so as to take advantage of the strengths of each. In this section, we describe an existing computing environment that does this. This environment is based on the software environment XploRe (<http://www.XploRe-stat.de>) but could also be implemented using any other system that provides the needed Java interface. We consider three uses of the client-server technology:

1. Instruction in econometrics via a Java interface and tutorials and interactive course texts.

2. Methods for enabling outside researchers to supply programs that other outside researchers can access over the Internet and use in applications.
3. The creation of Method and Data Technology centers for a group of suppliers

4.1. Instruction in Econometrics: In teaching econometrics, it is important to give students opportunities to experience the numerical properties of the methods they are using, develop intuition about how methods perform, and apply these methods to data. One way of doing this is to carry out instruction in a setting that has the required computers, software, and data – for example, a computerized classroom in which each student sits at a microcomputer that has access to the required application programs. This approach, while effective, can have the disadvantage of requiring much input of faculty time and effort to develop applications software and prepare data. Downloading software and ready-to-use data does not reduce these costs unless the software and data are compatible with the statistical software system being used for instruction.

The Java interface of XploRe offers the possibility of implementing econometric methods immediately and digesting formulas more easily by applying them to data directly. An example is given in a tutorial about generalized linear models (GLM) (<http://www.xplore-stat.de/tutorials/glmstart.html>).

The screenshot shows a Netscape browser window with the following content:

- Page Title:** Generalized Linear Models in XploRe - Netscape
- Location:** <http://www.xplore-stat.de/tutorials/glmstart.html>
- GLM Tutorial Contents:**
 - 1 Theory
 - 2 Data
 - 3 Macros
 - doglm
 - glmest
 - 4 Weights & Constraints
 - 5 Options
 - glmopt
 - 6 Statistics
 - glmstat
 - 7 Output
 - glmout
 - 8 Model
 - Selection
 - 9 Odds & Ends
- Buttons:** GLM Library, XploRe Home
- Main Content:**

Generalized Linear Models

by [Marlene Müller](mailto:marlene@wiwi.hu-berlin.de) <marlene@wiwi.hu-berlin.de>, November 19 - 1997

glnOutput

Edit Data Edit window Iconize

GLM fit, 'bilo', n=23

Estimates (b, s.e., t-value)

Parameter	Estimate	s.e.	t-value
b[1]	1.94469	0.3413	5.70
b[2]	1.12999	0.257	4.40
b[3]	-0.762634	0.2112	-3.61
b[4]	-0.847275	0.3222	-2.63
b[5]	0.227111	0.2501	0.91
b[6]	-0.736812	0.2989	-2.46

* constant variable: b[1]

This is a tutorial on how to use [XploRe](#) for estimating Generalized Linear Models (GLM). It basically describes the principal routines contained in the [glm](#) library.

The newest version of this document can be always found at:
<http://wotan.wiwi.hu-berlin.de/~marlene/tutorials/glmstart.html>

Figure 1. The GLM tutorial

It is assumed that the student has some theoretical background on the GLM. The tutorial instructs the student on how to apply what kind of model to what kind of data. The question of natural link functions is presented. The software may control a possible mismatch of data type and user chosen link by rejecting e.g. a logistic link for normal response data. The tutorial is written in HTML language. It is possible to inspect parts of the GLM module and, thereby, to introduce the student into the operational phase of applying the GLM technique to data. The Java interface may be opened via a simple mouse click on the user's desktop. This gives the student an independent computing window that allows him to apply GLM techniques to his own data. The computations are performed on a server. This may be the instructor's PC or workstation or a high speed remote server. The platform independent client window may be opened on any machine and any platform in the world. This enables the student to do his practical GLM course homework at home or even in an internet cafe.

The same technique is applicable to whole textbooks. A LATEX2HTML converter (<http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html>) may create an HTML document from a Latex text with links to the Java interface. Similar linking techniques to Java interfaces apply for PDF documents. Examples exist for books on Applied Multivariate Statistical Analysis (<http://141.20.100.247/~scripts/scripts/sma/ma.html>) and a course on Non- and Semiparametric Modelling (<http://141.20.100.247/~scripts/scripts/spm/spm.html>). A PDF document of a book on wavelets and their statistical applications is provided by (<http://141.20.100.247/~scripts/scripts/wav/wavpdf.pdf>). The course text is typically available as a PDF, Postscript or HTML document. The basic idea is that the student may carry out homework exercises without necessary physical access to the software environment used in class. Exercises and examinations may be carried out via email and HTML document transfer. Classes can be taught to a distant audience and no overhead emerges at these places since the Java interface is embedded into the course text. A student may write his own application in XploRe and make it available via the internet to other students or professors.

4.2 Supply of Technology: In writing an own application the user introduces in a sense a new technology that might be useful to add to the system of programs used for a specific course. This proliferation of technology among students and teachers is possible only if there is an easy and standardized way to "publish" user-written programs and macros. XploRe provides us with this technology. The student may easily produce a web visible HTML page of the results of his home work. Moreover he may add new algorithms and techniques to an existing faculty system of methods.

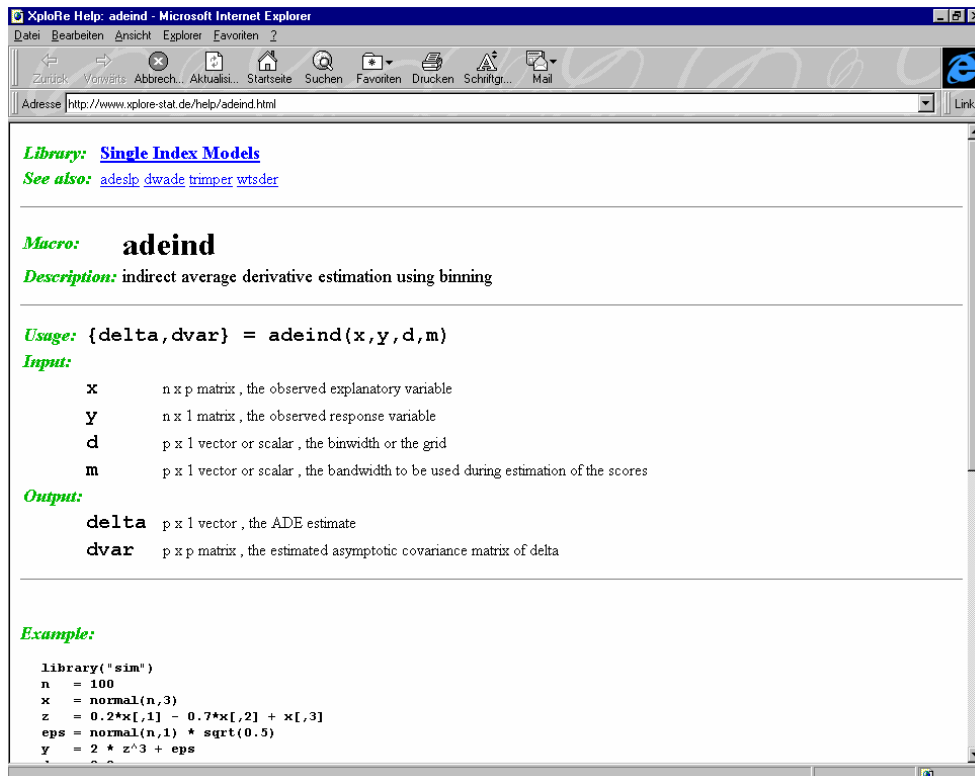


Figure 2: The help file for ADE made by a method supplier

The semiparametric technique for Average Derivative Estimation (ADE), for example, is hard to find in standard econometric packages, although much theoretical research has been carried out to investigate asymptotic properties. A researcher who has written a macro for ADE may put this technique on the web so that other researchers can try it with their own data. See the help page (<http://www.XploRe-stat.de/help/adeind.html>).

The Java interface, which may be started independently, serves as a client which takes contact with the server that is provided by the researcher, inventor and supplier of the help page. Many other examples of this kind, for example on ADE with discrete covariates, may be found on the help system pages (http://www.xplo-re-stat.de/help/_Xpl_Start.html). This technique leads in fact directly to a virtual computing and methodology laboratory.

4.3 MD*Tech Centers: The possibility of offering techniques to other researchers creates centers of technology which may be called Method and Data Technologies (MD*Tech) centers. They may provide outside users (professional clients, students, companies, etc.) with methodological techniques and data descriptions. An example would be a place that offers, say, high frequency finance data and fast forecasting methods. The clients may check the market with the methods that suit them but do not have to order a whole package of possibly unnecessary techniques. An example for methods of image processing is given by (<http://www.utdallas.edu/~degroat/javadip/JavaDIP.html>). Any user on the net may process several sample images via a Java interface. Buttons control the supplier written filtering and smoothing or false color imaging. A user's own data set may not

be introduced via the browser as explained above but a user who learns about a method from the web site may contact the supplier for further assistance.

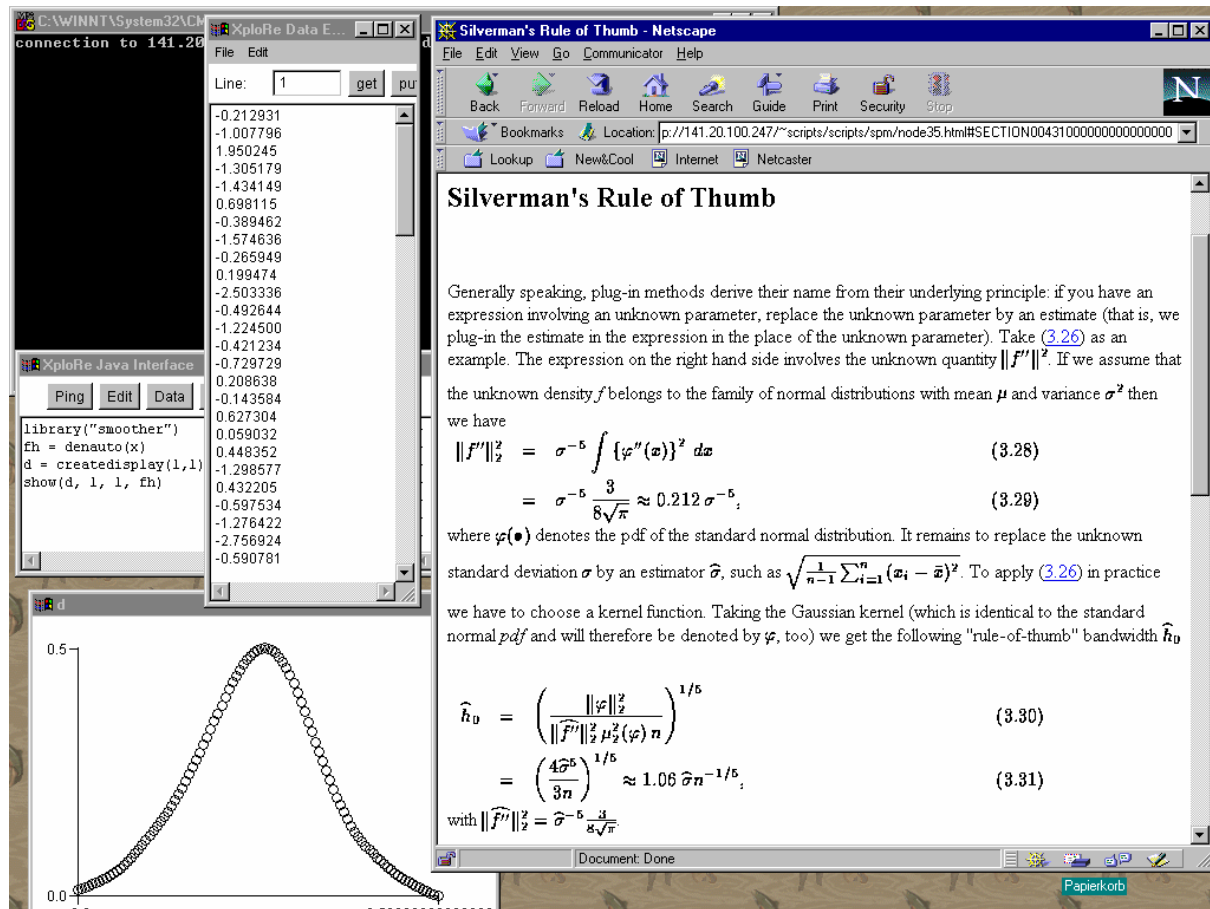


Figure 3: An instruction text on density estimation with a Java window

Figure 3 displays an example of an outside view on an MD*Tech center (<http://www.mdtech.de>). The client wants to apply a density estimation technique on his own data set. In the above mentioned course text on Non- and Semiparametric Modelling (<http://141.20.100.247/~scripts/scripts/spm/spm.html>) the method of automatic smoothing with Silverman's rule of thumb is described (right half of Figure 3). The student/user/client may now take his own data (upper left half of Figure 3) and apply the MD*Tech provided technique of automatic density estimation to his data. The resulting density estimate is provided as a graph in the lower left half of Figure 3.

Any individual or a research group may form such a MD*Tech center. The basic idea is that research oriented individuals or groups offer their knowledge and expertise via the internet together with a client/server based computing service. The service and methodology provider and the outside users of such a center may profit from this arrangement in two ways. The MD*Tech center that develops new methods (faculty, research group, etc.) is able to provide newest technology in shorter cycles than usual software updates. Second the outside user need not buy a whole software package on its own since the methods may be tailored for his needs. In addition, the Java interface reduces costs for both sides. The service provider may offer problem solutions for all platforms and, thereby, reduces programming work. The user may apply the provided methods to his data sets regardless of the user's platform. In

summary: this kind of internet based econometric computing can be of real value for clients and service providers. In the long run a MD*Tech center may even provide computing services for other institutions and thereby create a profitable marketing of university hosted technology.

5. Conclusions

Modern econometrics requires transparent use of highly specialized methods that are usually implemented on specific hard- and software platforms. The supply and proliferation of new econometric technology is complicated through this heterogeneity. Typically, software is written and optimized for a specific platform and thus not available for applied research at other places. The internet gives the potential for reducing this heterogeneity. We discuss possible software architectures for employing the potential of internet based econometric computing. With existing browser technology program pieces may be used in the world wide web with the exception of local file I/O.

A client/server concept encompasses this problem and offers the possibility of intra- and internet based teaching. We refer to existing examples of net-based teaching and present concrete examples for interactive teaching of elementary econometrics and statistics. For web based courses it is vital to have interaction between the clients data and the server provided methodology. The Java language is the appropriate tool for the architecture of this interaction. It allows platform independent computing via standard browsers. The net-based teaching may be seen as an export of technology to outside clients or students. This point of view leads to the potential for a group of teachers and researchers to form Method and Data Technology (MD*Tech) centers. In such MD*Tech centers methods for special econometric problems may be collected and via a Java interface applied researchers may use the provided methods with their own data.

Acknowledgements

We would like to thank Knut Bartels, Alan Kirman , Christian Müller, Marlene Müller, and Erich Neuwirth for helpful suggestions and corrections. The paper was financially supported by the Sonderforschungsbereich 373 "Quantifikation und Simulation Ökonomischer Prozesse", Deutsche Forschungsgemeinschaft. The research of Joel L. Horowitz was supported in part by NSF grant SBR-9617925.

References

- Applied Multivariate Statistical Analysis book (passwd protected)***
(<http://141.20.100.247/~scripts/scripts/sma/ma.html>)
- Average Derivative Estimation helpfile***
(<http://www.XploRe-stat.de/help/adeind.html>)
- CGI calculators***
(<http://www.stat.ucla.edu/calculators>).

CodEc software archive (<http://netec.mcc.ac.uk/CodEc.html>)

ELSA archive (<http://elsa.berkeley.edu/>)

GLM tutorial (<http://www.xplore-stat.de/tutorials/glmstart.html>)

GAUSS software (<http://www.aptech.com>)

GAUSS programming for Econometricians (<http://eclab.econ.pdx.edu/gpe/>)

Help system pages (http://www.xplore-stat.de/help/_Xpl_Start.html)

Image processing with Java (<http://www.utdallas.edu/~degroat/javadip/JavaDIP.html>)

LATEX2HTML converter (<http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/latex2html.html>)

MD*Tech – Method and Data Technologies (<http://www.mdtech.de>)

Non- and Semiparametric Modelling course text (passwd protected) (<http://141.20.100.247/~scripts/scripts/spm/spm.html>)

SAS software (<http://www.sas.com/>)

SHAZAM software (<http://shazam.econ.ubc.ca>)

Splus software (<http://www.mathsoft.com/Splus>)

Stata software (<http://www.stata.com>)

STATLIB server of SPLUS (<http://lib.stat.cmu.edu/S/>)

SticiGui© Java Tools (<http://www.stat.berkeley.edu/users/stark/Java/index.htm>)

SUN's Java Development Kit (JDK) (<http://www.javasoft.com/>).

Support Vector Machine (<http://svm.dcs.rhbnc.ac.uk/pagesnew/1D-Reg.shtml>)

Virtual Stat Lab (http://www.ruf.rice.edu/~lane/stat_sim/index.html)

wavelet book in PDF format (user=wave, pwd=let) (<http://141.20.100.247/~scripts/scripts/wav/wavpdf.pdf>)

Webstat Project (<http://www.stat.sc.edu/~west/webstat/>)

XLISP-STAT (http://www.cern.ch/WebMaker/examples/xlisp/www/cldoc_1.html)

XploRe – the internet interactive statistical computing environment (<http://www.XploRe-stat.de>)