# The R and Omegahat Projects in Statistical Computing

Brian D. Ripley

ripley@stats.ox.ac.uk
http://www.stats.ox.ac.uk/~ripley

# Outline

- Statistical Computing
  - History
  - S
  - R
- Application and Comparisons
  - Web servers
  - Embedding — Medieval Chant
  - R *vs* S-PLUS
- Omegahat and Component Systems
  - The Omegahat project
  - Components — GGobi
  - The future?

# Statistical Computing and S

# Scene-setting: Statistical Computing

1980

Mainly Fortran programming, or PL/I (SAS).
Batch computing (SAS, BMDP, SPSS, Genstat) with restricted range of platforms.

Some small interactive systems (GLIM 3.77, Minitab).

Very poor interactive graphics (2400 baud to a Tektronix storage tube if you were lucky). Flatbed and drum plotters, microfilm for publication-quality output off-line.

Mainly home-brew solutions in research. (GLIM macros?)

# 1990

PCs become widespread, but FPUs still uncommon. Sun etc workstations available for researchers, and for teaching in a few places.

Graphics could be pretty good (postscript printers, ca $1000 \times 1000$ pixel screens), but often was not, and mono text terminals were still widespread.

C was beginning to be used, as more portable than Fortran. (Few PC Fortran compilers then and now.)

Still SAS, SPSS etc as batch programs.

S beginning to be make an impact on research and teaching.

# 2001

Little spread in machine speed (min 500MHz, max 1.5GHz), fast FPUs are universal.

Colour everywhere, usually 24-bit colour.

The video-games generation is now at university.

Few people would dream of writing a complete program for a research idea: prototype and distribute in a higher-level language such as S or Matlab or Gauss or Ox or ....

Fortran is still used in scientific computing, but C or C++ is preferred, and Java has its advocates. SAS lives on as pseudo-batch program.

Lots of specialized tools are widespread, such as Perl, Python, Web browsers.

XML (eXtensible Markup Language) is the flavour of the year.

# Scene-setting: The ‘S’ Language

Largely the work of one person, Dr John M. Chambers of Bell Laboratories (formerly AT&T, now Lucent Technologies).

Awarded the prestigious 1998 *Association for Computing Machinery* Award for Software Systems for, in the words of the citation,

> *the S system, which has forever altered how people analyze, visualize, and manipulate data.*

For the last decade it has been the major vehicle for the delivery of new statistical methodology to end users.

S has a long history: the GR-Z graphics system goes back to 1976. JMC is now a Bell Labs Fellow, and is working on *Omegahat*, so that can be considered the successor to S.

# S History

The names have changed ('New S' and 'QPE' came and went) but the flavours of S are now known mainly by the colours of the covers of the books co-authored by Chambers.

| | | | |
|---|---|---|---|
| S1 | 1984 | *brown* | macro-based extension language |
| S2 | 1988 | *blue* | user-written extensions as first-class objects |
| S3 | 1991 | *white* | classes, some statistical functionality |
| S4 | 1998 | *green* | more rigorous class system |

All were Unix programs written in C and Fortran.

S-PLUS was first produced in 1988 by a start-up in Seattle called *Statistical Sciences* which in 1993 acquired exclusive marketing rights to S and merged with *MathSoft*. In 2001 they demerged and became *Insightful*.

S is not thought of by its developers as a statistical system, rather as *an interactive environment for data analysis and graphics*, a system within which to do statistics.

S-PLUS has been available for a limited range of Unix platforms and DOS and then Windows. It was not available for Linux until 1998, and never for Macintoshes.

The Unix versions have been based on S4 since 1998: all future Windows versions will be (now due August 2001).

S-PLUS is very widely used for teaching statistics at graduate level. Some of the early enthusiasts were earth scientists, and it has been used for service teaching. It has had less impact for mainstream undergraduate teaching, despite radical approaches like Nolan & Speed (2000) *Stat Labs: Mathematical Statistics through Applications*.

Academic licences for S-PLUS remain fairly expensive (although there is a CHEST deal in the UK). It is now pretty successful in several commercial sectors (finance, pharmaceuticals, manufacturing).

# What is R?

# R History

R is a system originally written by Ross Ihaka and Robert Gentleman of the University of Auckland (so the naming is clear) in the early 1990s. To the user it looks like a dialect of the S language, but the internal implementation is based on ideas from Scheme (a member of the LISP family). It is 'not unlike' S3.

Probably this started as a research project, but versions were used at Auckland for elementary classes, on Macintoshes with 2Mb of memory.

By 1997 other people had become involved, and a *core team* had been set up with write access to the source code. (No one kept records of who joined when.) There was a Windows version, and Linux users pushed development forward, there being no S-PLUS version available for Linux.

I became involved in 1998, and a member of the core team in Jan 1999.

The first non-beta version of R, 1.0.0, was released on 29 Feb 2000.
The latest, 1.2.3, was released on April 26th.

# Where is R now?

It is a system available as source code (at `www.r-project.org`) that compiles on almost all current Unix and Linux systems, and has binary versions for the major Linux distributions (Red Hat, SuSE, Debian, Mandrake), FreeBSD and 32-bit Windows and classic Macintosh (which also runs on MacOS X, on which the Unix port also builds).

It is distributed under GPL2 (the GNU Public Licence).

The core system is fairly small but can be extended by *packages*, 10 of which ship with R and over 100 are available (13 'recommended') from CRAN (`cran.r-project.org` and mirrors). Collectively these cover a wide range of statistical functionality, mainstream and oddball.

Most things one can do with S-PLUS can be done with R and its packages.

# Applications and Comparisons

# What is R being used for?

With a freely-distributable product, it is hard to know! However, users tend to ask for help, and a few contribute.

One of my main motivations for being involved is a (perhaps *the*) major use, to provide a first-class statistical system to students and researchers in the third world.

There are now many examples of R being used for large-scale data analysis. It was used for election forecasting in Austria and will be used (by David Firth) in the UK. My group use it to analyse 100Mb brain images.

There are several applications in gene expression arrays, at least two of which are commercial systems built on R, and one, `sma`, is available from CRAN.

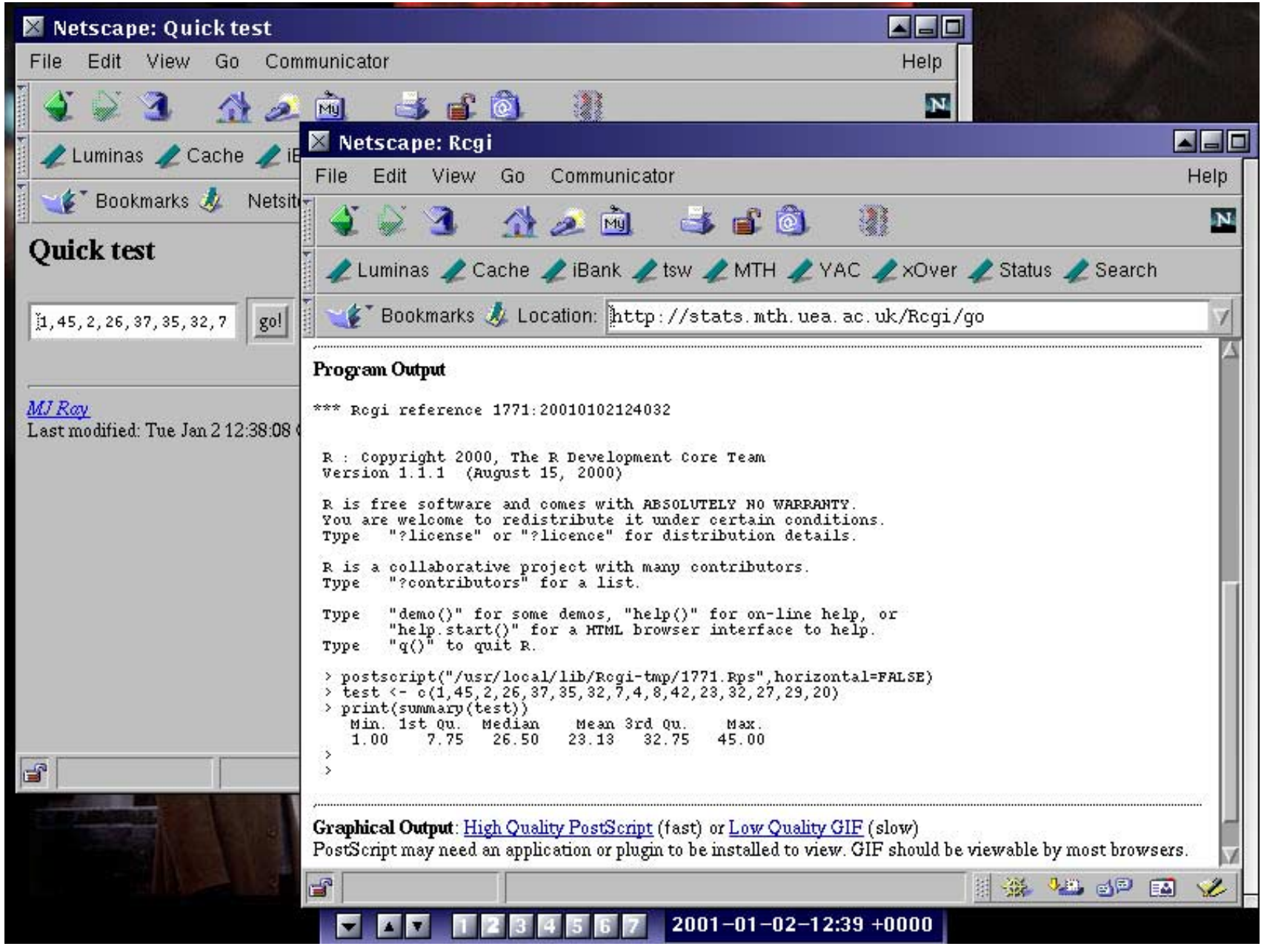
It is clear that researchers in many commercial companies are building systems around R.

# Web-based Statistical Teaching

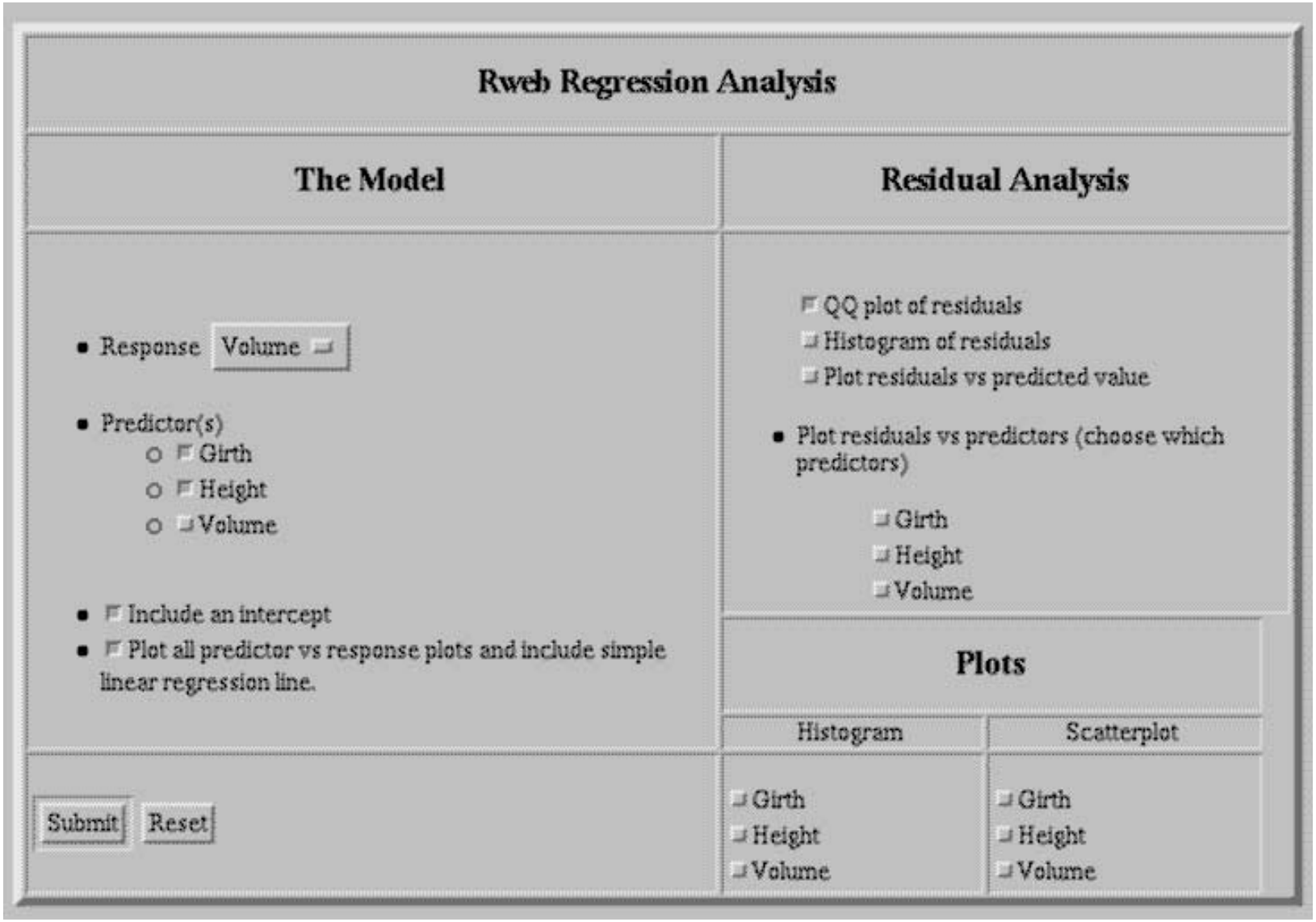
There are two harnesses, `Rcgi` (Mike Ray, UEA) and `Rweb` (Jeff Banfield, Montana State), to running R sessions from Web browsers. Both provide a simplified teaching interface.

`Rweb` provides 'a set of point and click modules that are useful for introductory statistics courses and require no knowledge of the R language'.

# Rcgi Example



# Rweb Example Module



# Embedding

Embedding can be taken much, much, further.

It is most advanced on Windows, where Thomas Baier's DCOM interface allows R to be called from Excel, Visual Basic, ..., but there is also a Unix/Linux version of R as a shared library.

These enable R to do what it does best, statistical analysis and presentation graphics.

# Medieval Chant

Musicologists undertaking detailed analyses of manuscripts of Western Christian liturgical chant dating back to the ninth century CE would welcome computer assistance. (Emma Hornby & John Caldwell, Faculty of Music in Oxford, statistics by Ruth Ripley.)

The early manuscripts employ several different notations, using *neumes* rather than notes. There are about twenty-five neumes, plus markings.

There a few thousand known chants with further variations between manuscripts. Ideally one would use optical character recognition to read them in, but exploring the feasibility of that is a project for a Master's student this summer.

At present chants are entered by a point-and-click data entry system written in Visual Basic.

# Medieval Chant: Design Issues

- The system has to be usable on fairly minimal Windows PCs by users whose experience stretches to Word and Internet Explorer.
- Need to build a database of chants.
- Non-trivial display issues: involved designing a TrueType font.
- The matching algorithms to be used are fairly complicated and subject to tweaking, and will result in a similarity matrix $S$.
- Given $S$, use standard multivariate techniques to compare chants (or verses or phrases of chants).

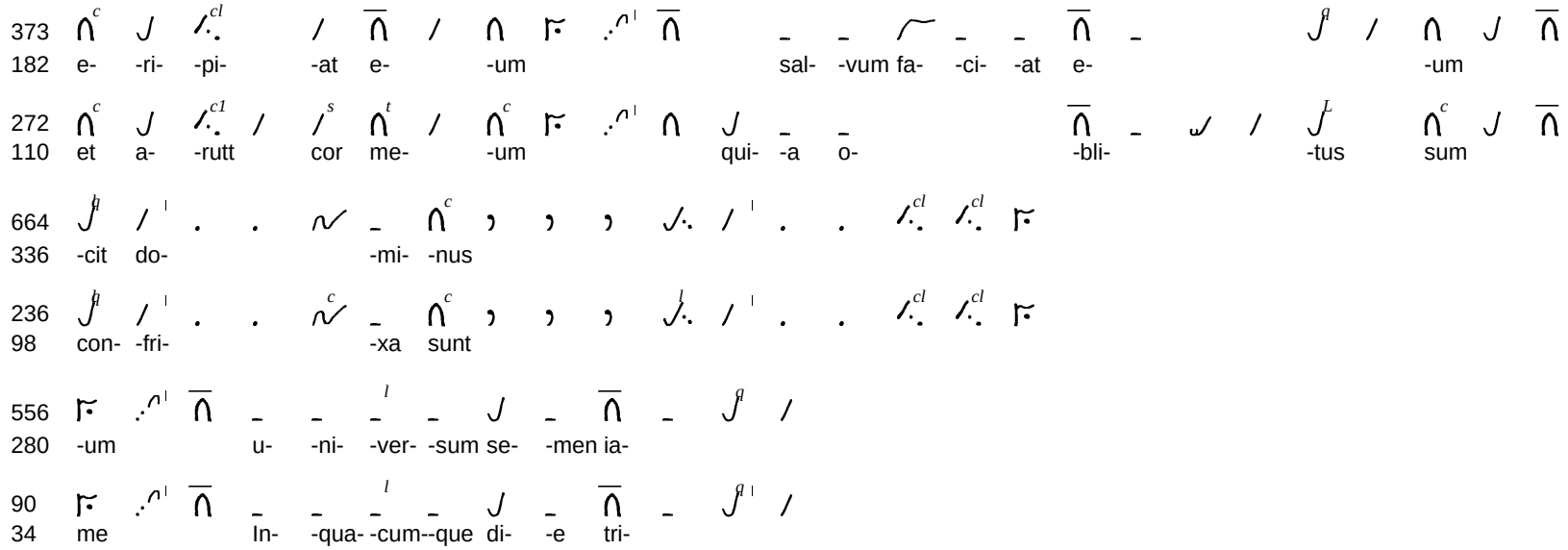
Solution has been to use a Visual Basic front-end driving a database interface and also a connection to an R server via DCOM.

# Medieval Chant: Sample Results

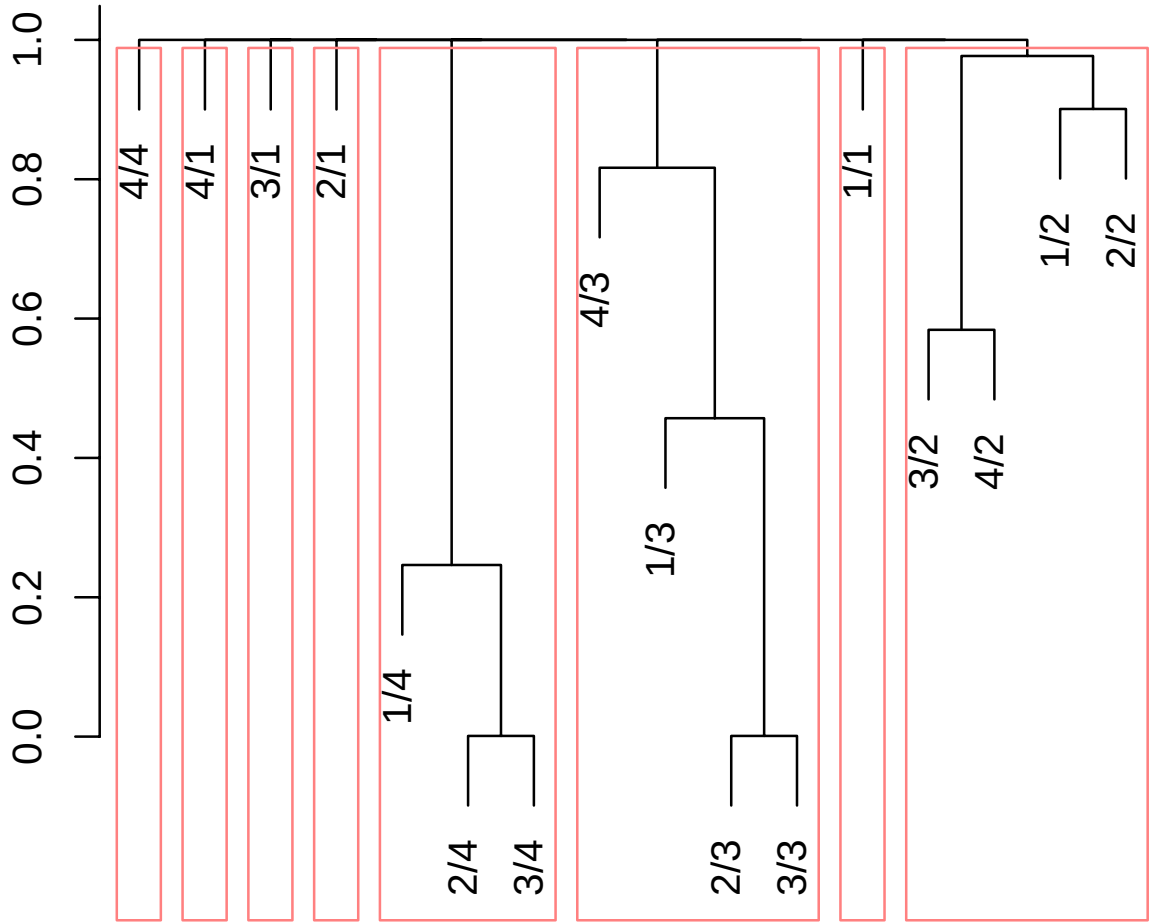
Similarity matrix for chants:

| | Deus deus meus | Domine exaudi | Audi filia | Domine audivi a |
|---|---|---|---|---|
| Deus deus meus | 00.0 | 26.0 | 28.0 | 29.0 |
| Domine exaudi | 00.0 | 00.0 | 25.0 | 41.5 |
| Audi filia | 00.0 | 00.0 | 00.0 | 21.0 |
| Domine audivi audit | 00.0 | 00.0 | 00.0 | 00.0 |

Deus deus meus compared with Domine exaudi : (similarity 26.00)



# Sample Results: Analysis




Dendrogram of phrases within the four verses of a chant, with groups highlighted.

# R *vs* S-PLUS

The two systems co-exist, uneasily at times.

- S-PLUS is commercial. R is freely distributable.
- R is much smaller and runs on less powerful machines. On Windows it fits on four floppies and (I'm told) runs on an 8Mb Windows 95 machine.
- S-PLUS is monolithic: R has a small core plus many extensions.
- S-PLUS on Windows has a 'menus and dialog boxes interface'. R does not, and although there are means to program one, in C, Tcl/Tk or Java, they are laborious. [Demos]
- Their performance is about equal, but R is much more tolerant of badly-written code that can make S-PLUS crawl.
- There is not much to choose in quality these days. I suspect R has more bugs, but they will be fixed faster by subject-matter experts.

- Both have 2D graphics of very high quality.
- R is currently missing the rich facilities of S for multi-panel plots (Trellis graphics) but prototypes have been demonstrated and this should appear in 2001.
- But both are poor on 3D graphics and dynamic and interactive graphics. S-PLUS on Windows is better than on Unix, and an add-on for R on Windows is under development. [`Rgl` by Duncan Murdoch: Demo]

It seems clear that the research emphasis in statistical computing has shifted from S to R: John Chambers is now a member of the R core team. The future looks like collaboration rather than competition.

# Working on an ‘Open Source’ Project

# ‘Open Source’ and ‘Free’ Software

These are emotive terms, coined by zealots.

Richard Stallman’s (RMS) Free Software Foundation is ‘free as in speech, not free as in beer’. The GNU project was set up to provide a ‘free’ Unix but made slow progress. In the early 1990s Linus Torvalds came along with the missing piece, a kernel, and *Linux* was born. However, well over half a ‘Linux distribution’ is from GNU, and RMS and others (the Debian distribution) refer to GNU-Linux.

The GNU Public Licence is more a manifesto than a licence, and deliberately contaminates. That is, you are allowed to use and distribute (modified versions of) GPL-ed software, but you can only *distribute* them as part of something else if that itself is GPL-ed (and if it were not, it becomes so).

There are other free licences (X, BSD, Apache, Artistic, . . . ), and the term ‘Open Source’ was coined for the concept, with a precise definition (by Bruce Perens). Some GNU-Linux distributions are purist about allowing only Open Source programs to be included.

These issues matter to some people.

Ross Ihaka has had battles with his University to be allowed to give away his work on R.

The right to build a commercial system on top of R is not clear.

The ownership of R's source is not very clear.

R makes use of RSS's Statistical Algorithms. We have a licence to do so on the understanding that this project is not commercial.

Probably one would not start from here!

# R is an Unusual Open Source Project

There is a much-quoted missive by Eric Raymond, 'The Cathedral and the Bazaar' about the merits of Open Source development.

It seems to me to be a romantic myth. Most of the successful Open Source projects were either commercially sponsored (X, Linux, GCC, ...) or dominated by a single leader.

R is really unusual in being a successful collaborative effort. It contrasts with XLISP-STAT, an Open Source system by Luke Tierney which has moved much more slowly. (Luke is now an active participant in R, and is setting up a core team for XLISP-STAT.)

R has no leader and operates by consensus in a core team of about 15 people. There are areas where certain members are regarded as expert, and there is a principle that those who will implement the ideas get more votes.

Ultimately we all respect Ross's views as the founder, though.

It seems it avoids being ‘a horse designed by a committee’ by

- Having a clear model, the S language, which was designed by one person.
- Being a loose confederation of areas designed by one person each (e.g. Ross Ihaka for the 2D graphics, Guido Masarotto for the Windows GUI, me for the external connectivity).
- Members being able to veto proposals which they see as inhibiting other developments (but this has been mis-used).

In general this works well, but there have been ‘robust’ debates. The team has only ever met twice (in Vienna in 1999 and 2001), and indeed most members have never seen Guido.

Ross put the criteria for core membership as ‘when it is more work to keep someone outside than invite them to join’.

# Problems

These projects are hard work, especially once they get real users.

- Users of free software are incredibly demanding. We say ‘R is a collaborative project with many contributors’ on the start-up banner, but in reality a tiny proportion of users ever contribute (less than 100 in total?)
  - They expect the system to build and work on their own system, and if it does not, they expect the core team to fix it.
  - They expect the system to behave the way they guess it should, and file a bug report rather than reading the documentation.
  - Many of the most demanding users are clearly using the system for commercial gain.

  One way forward may be to follow the Linux model of companies providing support to a free product.

- Backwards compatibility can become a millstone, as JMC found in going from S3 to S4. Users expect things they did in 1998 to work unchanged. It is likely that there will an R version 2 soon that is not backwards compatible.
- The developers get to know far more than they could possibly have wanted about the intricacies of operating systems and compilers.
  - We have found bugs in Linux C compiler, `gcc`, and lots in different Linux distributions.
  - We are continually working around design errors in `glibc`.
  - Adding date-time support became a nightmare. For example, Windows does not support dates before 1970-01-01, MacOS has no notion of time zones, and both Linux and DEC OSF-1 crash with correct examples of ISO C code.
  - Windows 2000 introduced a new set of incompatibilities with documented OS features (and they are still there).

# Successes

- R is widely used in the third world by groups who could never afford a commercial statistical system, and can be run fairly well on legacy hardware.
- We get to appreciate a lot more the design issues in statistical software, and the forbearance of people doing technical support.
- The core team have got to establish close working relationships with almost exclusively email contact. (Nothing new for me: Bill Venables and I co-wrote half a book before we ever met.)
- It is easy (for those who have made the effort to learn the system) to add new features whenever one's projects (or one's students' projects) need it.
- If you find a bug, just fix it and carry on.

Omegahat

# The Omegahat Language

Instead of S version 5, John Chambers and his colleague Duncan Temple Lang in 1998 started another track, a language called Omega. This arose out of discussion with (some of) the R developers and Luke Tierney on web-based software and distributed computing.

The *hat* was a contribution of Bill Venables to give a statistical flavour.

The Omegahat language (`www.omegahat.org`) is an interactive environment, effectively an interactive front-end to Java. It seems to have a handful of users.

The Omegahat project has also a range of Java packages implementing methods of interest in statistical applications.

# The Omegahat Project

The aims of the project are wider, to provide a vehicle for collaboration in statistical computing, particularly to provide re-usable components.

As such it includes the whole R core team, the S and XLISP-STAT developers and several other people with cognate interests.

There is also a range of inter-system interfaces, for example between R or S and Java (of course), Perl, Python, Xalan, Netscape and PostgreSQL.

The project is also an umbrella, at present only for one sub-project, a proposals on interfaces to DBMSs.

Omegahat uses a new-BSD licence that is more permissive than GPL2.

# Component-based Statistical Systems

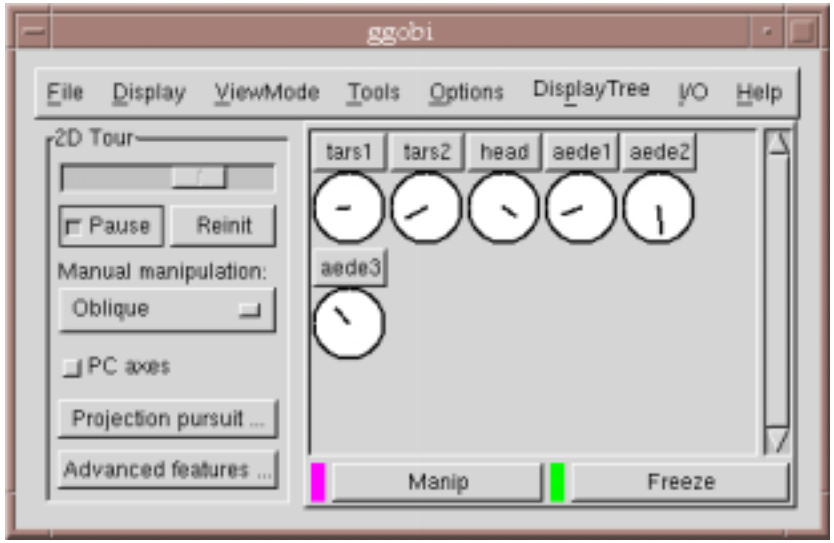
We have mentioned that R is lacking dynamic graphics. The obvious way to fill the gap is to borrow from another project, and Open Source licensing is intended to make such collaborations easy.

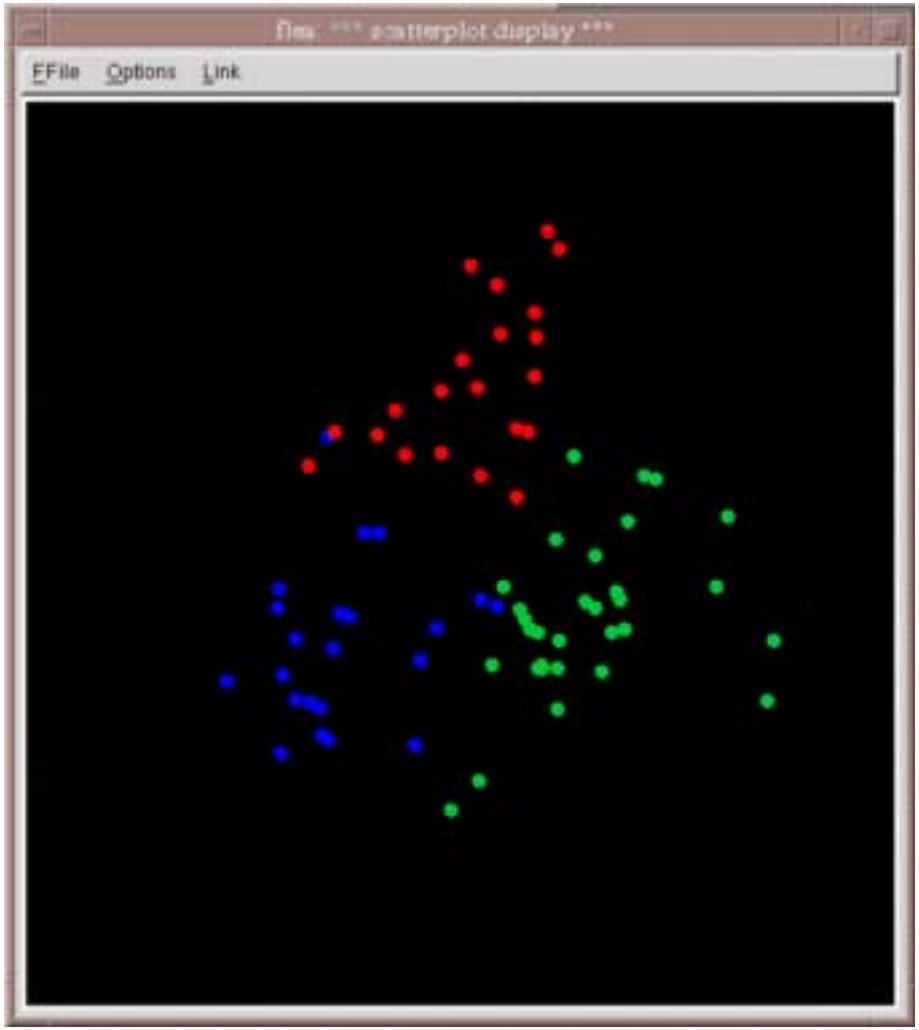
The XGobi system was developed at Bellcore/AT&T in the late 80's and early 1990's and provides an advanced dynamical graphics system. It runs under X Windows under Unix (and under Windows with an Xserver). Its developers were intending to move to the Gtk+ toolkit (developed for GIMP), and wanted a scripting language.

Out of this came GGogi (`www.ggobi.org`, still alpha/beta) which can run standalone (under Unix or Windows) or be embedded in R (or S-PLUS on Unix) which provides a scripting language.

The GGogi integration is currently the most successful, but the Perl and Python interfaces are usable. These present a two-way object-oriented view, so that Perl objects can be manipulated as if they were R objects in R, and *vice versa*.

# Ggobi ‘Grand Tour’ Example



# Where are R and Omegahat going?

This is hard to predict: volunteer projects go where the volunteers want to take them. But some predictions:

- The S language and (first) the R implementation will gain real object-oriented features.
- There will be a strong push towards establishing a synergy with DBMSs.
- There will be moves to ‘literate data analysis’ with XML as the glue.
- Threads, events, exceptions, . . . , will at last be tackled seriously.
- Cross-platform support will continue to be taken seriously.