May, 2006

Mathematica 5.2: A review

Joseph Hilbe, *Arizona State University*

Mathematica 5.2: A Review

Joseph M. HILBE

Mathematica is characterized by Wolfram Research, manufacturers of the software, as a "technical computing system." Most of those who use it, however, likely think of it as a comprehensive numeric and symbolic computational package with extensive associated graphical capabilities and as a programming language with an interactive document or notebook interface. Regardless of how the package is characterized, Mathematica is perhaps the most well-known and well-used technical computing package on the market. Current estimates show that Mathematica has a user base exceeding that of its two closest competitors—Maple and MathCAD.

The purpose of this review is to provide the reader with an overview of Mathematica 5.2's scope and capabilities, its limitations and any points needing development, and a sense of how the Mathematica program actually works. To achieve the latter end, I shall provide several worked out examples demonstrating the package at work. Given the fact that the Notebook interface can be saved in LaTeX format, replicating actual example input and output should clearly represent how the screen display appears.

I shall first provide details of license costs, together with a listing of what comes with the package. Following this I'll provide a brief history of Wolfram Research and of the evolution of the software. Next I'll list some of the most important enhancements that appear in version 5.2, which was released in July 2005. Thereafter I'll discuss the general capabilities of the package, followed by various examples. Finally, I'll mention any shortcomings or features of the package that I think need further development.

1. FEES AND SUPPORTING PRODUCTS

Mathematica has been designed to work on a number of platforms. This review relates only to personal computer implementations, including MS Windows, Macintosh, and Linux systems. Version 5.2 single user licensing costs, in U.S. dollars in the United States and Canada, for these platforms are:

Version	*Fee*	*(upgrade from ver 5.1)*
Professional	\$1880	(\$375)
Government	\$1580	(\$375)
Academic (4 yr college)	\$895	(\$225)
Student (all levels)	\$140	(\$100)

Semester and annual versions are also available for the student version starting at \$45. The cost for professional and government licenses includes one year of premier service. In addition, Wolfram publishes a number of ancillary packages that rely on the Mathematica kernel for number and symbolic calculations. Depending on the package, many of the capabilities of the full Mathematica have been dropped. Four of the many separate packages include:

Teacher's Edition (precollege and CC) version 1, \$195. The toolkit interface allows a teacher to generate customized assignments and quizzes together with their answer keys, construct animated demos and interactive courseware, and format printed results using mathematic typesetting. The Teacher's Edition also comes with a calculator system.

Mathematica CalcCenter, version 3, (\$595) and Mathematica CalcCenter for Students (\$99.95). The CalcCenter provides a number of palettes that allow users to solve problems without directly using a command interface. It has nearly the same numeric, and only marginally diminished symbolic, calculational capability as does Professional Mathematica. It cannot deal with symbolic differential equations, or more complex mathematics. Other limitations exist as well. But it does allow the user to engage in data analysis, descriptive statistics, plotting, and technical report writing. I intend on reviewing this package at a later date.

MathStatica 1.5 (\$159). Created by a cooperating vendor, MathStatica uses the Mathematica kernel to provide the user with the capability of working with a wide variety of probability distributions and associated random variables, with engaging in maximum likelihood estimation, and calculating a number of statistics that are normally found in commercial statistical packages. A text, entitled *Mathematical Statistics with Mathematica* (Rose and Smith 2002), comes with the package. I shall also be reviewing MathStatica separately in a future issue of *TAS*.

Statistical Inference Package, version 1. The package, just released, has been specifically designed to make it easy for users to do classical likelihood-based statistical inference. Capabilities include procedures for maximum likelihood estimation, for example, regression models for univariate and multivariate linear and nonlinear models (examples: logistic, Poisson multinomial), profile-likelihood-based confidence intervals for parametric functions, likelihood ratio tests, and the ability to form complicated models from simpler ones. I intend to review this package in a subsequent issue of *TAS*.

webMathematica Professional, version 2, (pricing varies, contact sales) and webMathematica Amateur (available for free to Premier Service subscribers). webMathematica enables users to create Web pages that access Mathematica's computational engine for specific applications. The Web pages could, for instance, include combinations of symbolic and numeric math along with relevant graphics.

A free webMathematica program, called *The Integrator*, can be accessed at *http://integrals.wolfram.com/index.jsp*.

The Integrator uses webMathematica technology to provide an interface whereby a user simply enters the formula to be inte-

Joseph M. Hilbe is Software Reviews Editor for *The American Statistician* and has the dual position of Professor Emeritus, University of Hawaii, and Adjunct Professor of Sociology and Statistics, Arizona State University, Tempe, AZ (E-mail: *jhilbe@aol.com*). I wish to thank both Ben Wilson, Media Relations Manager, Wolfram Research, for sending me materials and references used for this review, and Darren Glosemeyer, Wolfram Research for his excellent help with the technical aspects of the package, and for assuring that I did not misrepresent its capabilities. I also thank the referees who provided useful comments and suggestions.

 DOI: 10.1198/000313006X110483

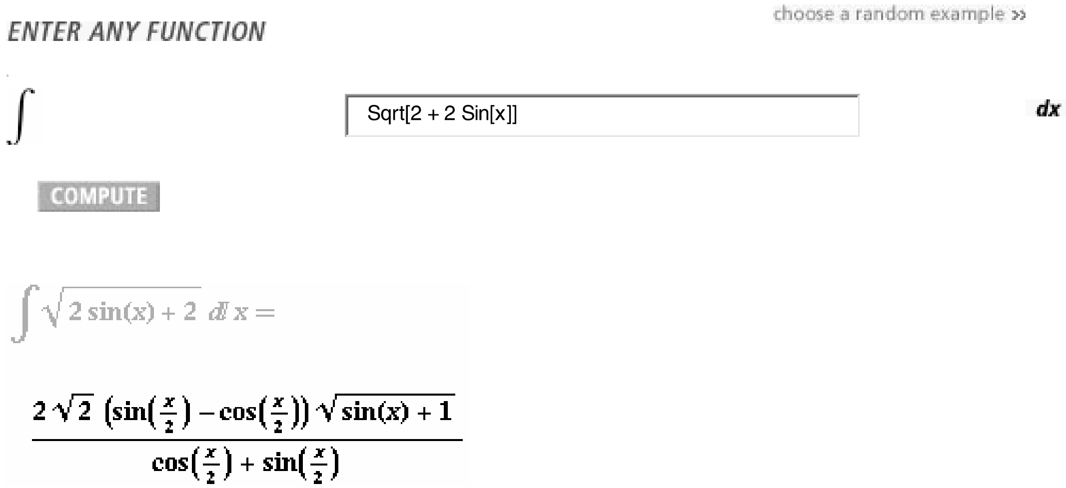


Figure 1. Using the Integrator.

grated into the Integrator box and clicks on a button called "Compute" to compute an indefinite integral with respect to x. Figure 1 shows a sample session, where I typed "`Sqrt[2+2Sin[x]]`" into the box. The result displays a nicely formatted integration formula together with the result. I would have loved to have had this program when I first took calculus—to check answers of course!

You may learn more about theses packages by accessing *http://documents.wolfram.com/*.

Mathematica offers the user a number of resource Web sites. I recommend that the reader look at *http://www.wolfram.com/webresources.html*, which provides a menu to a variety of resources. For example, from the above Web site one can click on *functions.wolfram.com* and have access to 87,160 functions (as of February 16, 2006) on the Wolfram Functions Web site. Other sites of particular interest to statisticians are: MathWorld, an online mathematics encyclopedia at mathworld.wolfram.com with Mathematica Notebooks for many entries, and the Wolfram Information Center library.wolfram.com which contains programs, articles and conference proceedings for Mathematica and other Wolfram products.

Wolfram offers a number of additional domain specific specialization packages, each of which uses the built-in Mathematica computational kernel. Some of the more interesting include:

Advanced numerical methods	Control systems professional
Time series	Wavelet explorer
Experimental data processing	Neural networks
Digital imaging processing	Fuzzy logic
Mechanical systems	Parallel computing toolkit
Scientific astronomer	Structural mechanics

Additionally, Wolfram Research offers a number of on-site and Web-based courses on how to use Mathematica, as well as how to use Mathematica for particular interests. Of likely interest to *TAS* readers, Wolfram offers a two-part course on using Mathematica for statistical analysis. Each part is offered online as a half-day course at a cost of $250. The two parts are also offered together as a full-day course at training centers at a cost of $495. A discount of 30% is available to faculty and students for online and training center based courses. Companies can also contact the sales department to arrange special on-site training for their company. The Web site address for information regarding the "Statistics with Mathematica" course is: *http://www.wolfram.com/services/education/minicourses/m215.html.*

Links to other educational pages, as well as to the currently scheduled courses, are: *http://www.wolfram.com/services/education/* and *http://www.wolfram.com/services/education/calendar.cgi*.

2. WOLFRAM RESEARCH AND MATHEMATICA

Wolfram Research, Inc., founded in 1987 by Stephen Wolfram and located in Champaign, IL, is one of four companies owned by the Wolfram Group. Others include Wolfram Media, Inc., Wolfram Research Europe Ltd (UK), and Wolfram Research Asia Ltd (Japan). Wolfram still heads the privately held corporation, and oversees the development of Mathematica and all of its derivative products.

Wolfram, who received a Ph.D. in theoretical physics from CalTech in 1979 at the age of 20, began working on the construction of a computer algebra system, called SMP, in the same year. He finished it two years later in 1981, releasing it on the commercial market. After engaging in substantial work in the field of complex systems research, Wolfram began work on what was to become Mathematica in late 1986, while a professor of physics, mathematics, and computer science at the University of Illinois. He founded Wolfram Research a year later, and released the first version of Mathematica in June of 1988.

Since the earliest stages of computing, in particular personal computing, packages were marketed for algebraic computing, graphical presentation of data, and a variety of other numeric tasks. Wolfram sought to provide Mathematica with a new type of symbolic computer language that would allow for the defining and manipulation of a wide range of objects based on relatively few constants and primitive functions. Instead of the modular approach used by most other technical packages, Mathematica is a truly integrated system based on a hierarchical logic. What this brings the user will be apparent in the examples shown later.

Mathematica has undergone five major versions, with version 2 being released in 1991, version 3 in 1996, version 4 in 1999, and version 5.0 in 2003. The current version 5.2 was released in 2005. It appears that new major enhancements to Mathematica come at intervals of three to four years.

At first Mathematica was used primarily by those in the physical sciences, mathematics, and engineering. Over the some 18 years of its existence, users now come from every discipline having a reason to engage in numeric and symbolic calculation. However, its foremost clientele still appears to be those in the engineering and mathematical fields—and, of course, in education.

3. PACKAGE SPECIFICS

Mathematica comes with the following:

- Mathematica 5.2 CD
- *Getting Started with Mathematica 5.2.* (64 pages)
- *A Quick Tour of Mathematica 5.* (21 pages)

The Mathematica Book (Wolfram 2003), is located in the Help Browser and at *documents.wolfram.com*. At some 1,500 pages in length, the text provides the user with a complete guide on how to use the software. Numerous examples are given throughout the book. Extensive help is available from within the Mathematica package, as well as from both technical support and Web support. I contacted technical support without advising them that I was reviewing the package. The person answering my questions was extremely helpful and professional in his response.

3.1 System Capabilities and Limitations

Mathematica 5.2 requires 128MB RAM to operate, although the vendor recommends 256MB RAM or more. Additionally, the software takes up between 400 to 550 MB of hard disk space, depending on the options installed. These requirements are well within the capabilities of the majority of PC's and Macs currently on the market.

Mathematica 5.2 supports 64-bit memory addressing and 64-bit long number partitioning. This means that the memory limit is 2^{64} bytes or some 18 billion Gigabytes, well beyond current hardware limitations. Mathematica is the first desktop computational package with this capability.

4. MATHEMATICA 5.2: CAPABILITIES

In general terms, Mathematica 5.2 provides the user with a wide range of computational, graphical, and document creation capabilities. Employing a huge collection of computational algorithms, the user is able to perform a widely diverse number of complex numeric and symbolic calculations. Mathematica's built-in programming language and computational kernel can be used to create customized systems of algorithms.

The built-in numeric and symbolic functions make it possible to model and simulate a wide variety of situations, including complex biological systems, financial derivatives, environmental impact studies, health care economic impact alternatives, meteorological profiles, and even alternative consequences of galactic collisions. Built-in graphics functions make it possible to construct dynamic animated three-dimensional visual presentations, as well as a host of associated plots and graphs for data and functions.

The Notebook interface can also be configured to use a Mathematica kernel on another machine. In other words, one can use a laptop to enter data and prepare documents while using a remote computer for numeric and symbolic calculation.

Mathematica programs can be written in text files and run in Mathematica. In addition to accessing the kernel through Mathematica's Notebook interface, Mathematica can be run in a command line mode. Mathematica can also be interfaced with other programming languages. MathLink is the means by which Mathematica's Notebook interface and kernel interact. MathLink also provides an interface to programs in other languages, such as C. MathLink applications called J/Link and .Net/Link are included with Mathematica. These two applications provide integration of Mathematica with Java and .Net, respectively.

Capabilities in several areas of specific interest to statisticians include the following:

a. Built-in statistics functionality includes descriptive univariate and multivariate statistics, linear and nonlinear regression, ANOVA, cluster analysis, numerous univariate and multivariate distributions, and statistical plots. The regression functions include a number of estimates and diagnostics including parameter estimates, standard errors and confidence intervals and ellipsoids, goodness-of-fit measures, and leverage diagnostics. The distributions include random number generation, and symbolic and numeric results such as the mean, variance, skewness, kurtosis, pdf, cdf, characteristic function, and expected values. Various statistical plots built into the software include box-and-whisker, Pareto, pair-wise scatterplots, quantile plots, histograms, pie charts and various bar charts. Stem-and-leaf plots are new to version 5.2.

b. A large set of special functions is included in Mathematica ranging from elementary functions such as trigonometric, hyperbolic trigonometric and exponential functions to orthogonal polynomials and functions related to factorial, hypergeometric, elliptic and zeta functions. A few of the built-in special functions that appear often in statistics are the binomial, multinomial, gamma, beta, regularized gamma and beta functions, log-gamma, poly-gamma, multiple erf functions, generalized and confluent hypergeometric functions, Bessel, Legendre, Chebyshev, and Hermite polynomials, the unit step function and multiple delta functions. Mathematica can evaluate these functions both symbolically and numerically. Built-in formula manipulation functions and simplifiers are also included, which in many instances allows for the reduction of complicated symbolic expressions to simpler ones. I have found this feature to be particularly useful.

c. Data can be imported into and exported from Mathematica in many formats. The formats include common formats for text, numeric data, graphics, XML, HTML, binary data, and audio formats. A few of the supported text and numeric data formats include plain text, tab- or comma-separated formats, and Excel spreadsheets. A number of field specific formats are also included.

d. A DatabaseLink package is included with Mathematica. The package allows users to integrate Mathematica with database managements. DatabaseLink is fully JDBC-compliant. It contains both a SQL and Mathematica interface/

e. There are many types of numeric functions in Mathematica. A few of the numeric functions of special interest to statisticians are linear algebra functions, local and global optimization functions, and numeric root finding functions. The set of linear algebra functions is extensive, ranging from more basic operations such as dot product, transpose, and inverse to matrix decompositions including singular value, Cholesky and QR decompositions. Most of the linear algebra functions work for

Examples of Functions Related to Statistics; Finding Moments; Standard OLS Regression

<< Statistics`

Here are a couple of familiar limits for Student's t distribution: standard normal as n→ ∞, and Cauchy as n→1.

Limit[PDF[StudentTDistribution[n], x], n → Infinity]

$\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$

PDF[NormalDistribution[0, 1], x]

$\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$

Limit[PDF[StudentTDistribution[n], x], n → 1]

$\frac{1}{\pi+\pi x^2}$

PDF[CauchyDistribution[0, 1], x]

$\frac{1}{\pi(1+x^2)}$

Here is the expected value of sin(x) when x follows a normal distribution.

ExpectedValue[Sin[x], NormalDistribution[μ, σ], x]

$e^{-\frac{\sigma^2}{2}}\mathrm{Sin}[\mu]$

ExpectedValue calls Integrate to get expectations. Here is how the expectation could be computed directly.

Integrate[Sin[x] ∗ PDF[NormalDistribution[μ, σ], x], {x, −Infinity, Infinity}, Assumptions → {μ ∈ Reals, σ > 0}]

$e^{-\frac{\sigma^2}{2}}\mathrm{Sin}[\mu]$

This is an example of a symbolic evaluation where moments are obtained from characteristic functions and compared with direct integration results. For instance, take the characteristic function for χ^2_n.

charfun = CharacteristicFunction[ChiSquareDistribution[n], t]

$(1-2it)^{-n/2}$

Compute the first five raw moments by performing derivatives, taking the limit as t→0, and dividing by i^m.

Table[Limit[D[charfun, {t, m}], t → 0]/I^m, {m, 1, 5}]

$\{n, n(2+n), n(2+n)(4+n), n(2+n)(4+n)(6+n), n(2+n)(4+n)(6+n)(8+n)\}$

As a comparison, we can find the general form of the ith raw moment for χ^2_n.

rawmoments = ExpectedValue[x^i, ChiSquareDistribution[n], x, Assumptions → {i > 0}]

$\frac{2^i\mathrm{Gamma}\left[i+\frac{n}{2}\right]}{\mathrm{Gamma}\left[\frac{n}{2}\right]}$

Then, substitute the values 1 through 5 into the expression to get the raw moments.

rawmoments/.i → Range[5]

$\left\{\frac{2\mathrm{Gamma}\left[1+\frac{n}{2}\right]}{\mathrm{Gamma}\left[\frac{n}{2}\right]}, \frac{4\mathrm{Gamma}\left[2+\frac{n}{2}\right]}{\mathrm{Gamma}\left[\frac{n}{2}\right]}, \frac{8\mathrm{Gamma}\left[3+\frac{n}{2}\right]}{\mathrm{Gamma}\left[\frac{n}{2}\right]}, \frac{16\mathrm{Gamma}\left[4+\frac{n}{2}\right]}{\mathrm{Gamma}\left[\frac{n}{2}\right]}, \frac{32\mathrm{Gamma}\left[5+\frac{n}{2}\right]}{\mathrm{Gamma}\left[\frac{n}{2}\right]}\right\}$

It may not be immediately obvious that these Gamma ratios are the same as the moments obtained from the characteristic function above. Simplifying with the positivity assumption on n shows that the Gamma ratios are the same polynomials in n.

FullSimplify[rawmoments/.i → Range[5], Assumptions → n > 0]

$\{n, n(2+n), n(2+n)(4+n), n(2+n)(4+n)(6+n), n(2+n)(4+n)(6+n)(8+n)\}$

Figure 2(a). Some basic statistics—regression.

Regression

To demonstrate how *Mathematica*'s built-in regression functions work, I shall look at a simple linear regression. Nonlinear regression functions work in a similar manner. The following defines a data set.

```
data = {{5.31, 22.24}, {0.89, 7.67}, {5.69, 21.16}, {7.01, 33.76}, {3., 12.89},
{7.41, 37.22}, {3.46, 19.47}, {9.64, 39.83}, {2.46, 10.68}, {4.2, 25.38},
{6.56, 31.58}, {2.48, 17.15}, {3.89, 14.77}, {1.91, 11.37}, {7.8, 38.71}};
```

The semicolon after the input is used to suppress output. This is useful for instance if an output is expected to be large, or in intermediate calculations where the result is of interest only as an input for another operation.

The Regress function performs linear regressions and returns a few commonly used results by default.

```
Regress [data, {x}, x]
```

{ParameterTable →

	Estimate	SE	TStat	PValue
1	3.18353	1.94727	1.63487	0.126047
x	4.12951	0.362933	11.3782	3.94563×10^{-8}

,
RSquared → 0.908748, AdjustedRSquared → 0.901728, EstimatedVariance → 11.7215,

ANOVATable →

	DF	SumOfSq	MeanSq	FRatio	PValue
Model	1	1517.5	1517.5	129.462	3.94563×10^{-8}
Error	13	152.38	11.7215		
Total	14	1669.88			

}

A number of other diagnostics are included. The list can be obtained using the RegressionReportValues function.

```
RegressionReportValues[Regress]
```

{AdjustedRSquared, ANOVATable, BestFit, BestFitParameters, BestFitParametersDelta, CatcherMatrix, CoefficientOfVariation, CookD, CorrelationMatrix, CovarianceMatrix, CovarianceMatrixDetRatio, DurbinWatsonD, EigenstructureTable, EstimatedVariance, FitResiduals, HatDiagonal, JackknifedVariance, MeanPredictionCITable, ParameterCITable, ParameterConfidenceRegion, ParameterTable, PartialSumOfSquares, PredictedResponse, PredictedResponseDelta, RSquared, SequentialSumOfSquares, SinglePredictionCITable, StandardizedResiduals, StudentizedResiduals, SummaryReport, VarianceInflation}

These values are specified via the RegressionReport option. Replacement rules can be used to extract the values. For instance, here the fitted curve is extracted and assigned to the symbol fitted.

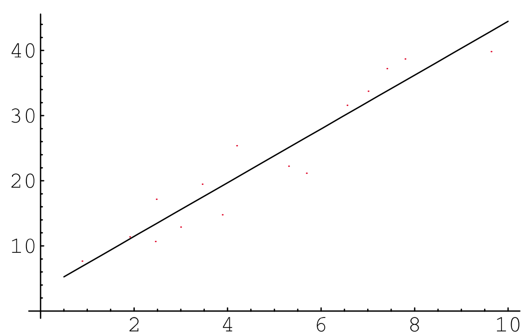
```
fitted = BestFit/.Regress[data, {x}, x, RegressionReport → BestFit]
```

$3.18353 + 4.12951x$

For a qualitative look at the fit, one might plot the fitted curve and the data in one graphic. Plot creates graphics of functions, ListPlot plots data, and DisplayTogether can be used to combine graphics.

```
<< Graphics`

DisplayTogether[Plot[fitted, {x, .5, 10}],
ListPlot[data, PlotStyle → {Red, PointSize[.012]}]]
```



−Graphics−

Figure 2(b). Some basic statistics—regression.

Poisson and Beta Estimation simulated variates graphical representations

```
<< Statistics`
```

Poisson Example

This generates 50 random Poisson variates. The semicolon inhibits display of output.

```
poissondata = RandomArray[PoissonDistribution[7.3], 50];
```

The loglikelihood function can be constructed by computing the pdf for a Poisson with mean μ at each of the data values, and summing the logs of those pdf values.

```
poissonllfun[μ_] = Total[PowerExpand[Log[Map[PDF[PoissonDistribution[μ], #]&, poissondata]]]]
```

$-50\mu - 5\mathrm{Log}[6] - 4\mathrm{Log}[24] - 5\mathrm{Log}[120] - 6\mathrm{Log}[720] - 10\mathrm{Log}[5040] - 3\mathrm{Log}[40320] - 3\mathrm{Log}[362880] - 9\mathrm{Log}[3628800] - 2\mathrm{Log}[39916800] - \mathrm{Log}[87178291200] + 339\mathrm{Log}[\mu]$

The maximum likelihood estimate of μ can be obtained in several ways. FindMaximum uses numeric methods and returns the maximized loglikelihood value as well as the value of μ at which it is maximized.

```
FindMaximum[poissonllfun[μ], {μ, 10}]
```

$\{-126.852, \{\mu \to 6.78\}\}$

In this example, μ can be computed by taking the derivative and solving for μ when the derivative is 0.

```
Solve[D[poissonllfun[μ], μ] == 0, {μ}]
```

$\{\{\mu \to \frac{339}{50}\}\}$

In this case, we could just note that the mean of the data values will give the maximum likelihood estimate of μ.

```
Mean[poissondata]
```

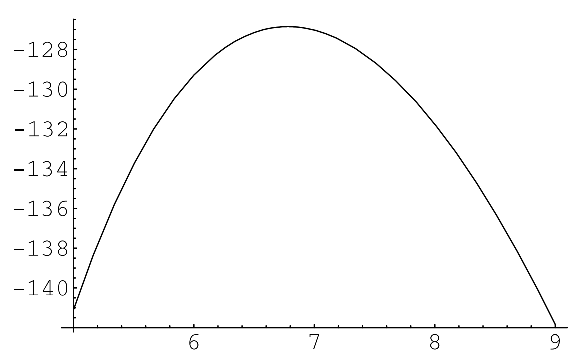
$\frac{339}{50}$

```
N[%]
```

6.78

Here is a plot of the loglikelihood function.

```
Plot[poissonllfun[μ], {μ, 5, 9}]
```



−Graphics−

Beta Example

Maximum likelihood for distributions with two parameters make for nice examples because they lend themselves to contour and 3D graphics.

This generates 100 random beta variates.

```
betadata = RandomArray[BetaDistribution[2.7, 1.4], 100];
```

Figure 3(a). Poisson and beta models.

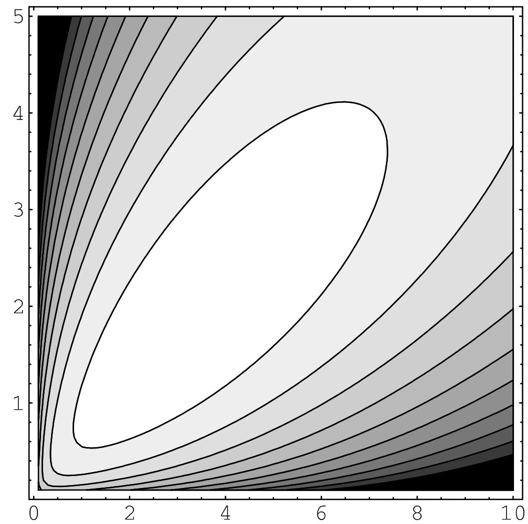
The loglikelihood function can be computed as in the Poisson case.

betallfun[alpha_, beta_] =
Total[PowerExpand[Log[Map[PDF[BetaDistribution[α, β], #]&, betadata]]]]

$-52.4047(-1+\alpha) - 121.204(-1+\beta) - 100\text{Log}[\text{Beta}[\alpha, \beta]]$

Here is a contour plot of the beta loglikelihood function.

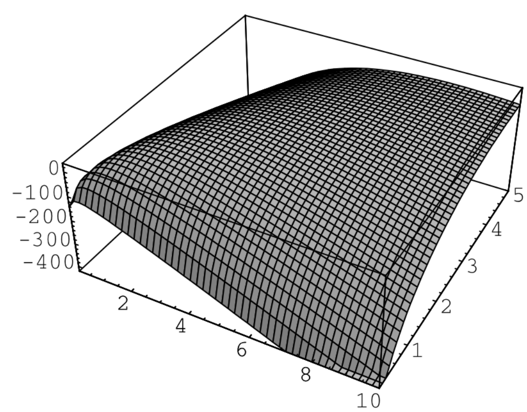
ContourPlot[betallfun[α, β], {α, .1, 10}, {β, .1, 5}, PlotPoints → 100]



−ContourGraphics−

This is an analogous 3D plot of the loglikelihood function.

Plot3D[betallfun[α, β], {α, .1, 10}, {β, .1, 5}, PlotPoints → 50]



−SurfaceGraphics−

The likelihood will be maximum when the following equalities are met.

derivs = {D[betallfun[α, β], α] == 0, D[betallfun[α, β], β] == 0}

{$-52.4047 - 100(\text{PolyGamma}[0, \alpha] - \text{PolyGamma}[0, \alpha+\beta]) == 0$,

$-121.204 - 100(\text{PolyGamma}[0, \beta] - \text{PolyGamma}[0, \alpha+\beta]) == 0$}

Numerical methods will be needed for this example. Here a solution to these equations is found by FindRoot.

FindRoot[derivs, {α, 5}, {β, 5}]

{$\alpha \to 2.96282, \beta \to 1.71452$}

As in the Poisson example, the maximum likelihood estimates can also be found using FindMaximum.

FindMaximum[betallfun[α, β], {α, 5}, {β, 5}]

{24.2845, {$\alpha \to 2.96282, \beta \to 1.71452$}}

Figure 3(b). Poisson and beta models.

Poisson Fitting

First, load the Statistics packages.

```
<< Statistics`
```

The following imports an Excel spreadsheet of data.

```
data = Import["mpar250.xls", "XLS"];
```

The variable names are contained in the first entry in the data and stored as strings.

```
data[[1]]
```

{los, hmo, white, died}

The variable los will be modelled as a function of hmo, white, and died. For the examples that follow, only the predictor variables will need names. The variables will also need to be symbols. The Rest function takes all but the first element of a list. The ToExpression converts strings to *Mathematica* expressions, so the following gives the list of variable names as symbols.

```
vars = Map[ToExpression, Rest[data[[1]]]]
```

{hmo, white, died}

In the following, Rest is used to drop the row of variable names from the data. Fitting functions in *Mathematica* contain the response as the last column. Transpose and RotateLeft are used to move the first column to the last column of data.

```
data = Transpose[RotateLeft[Transpose[Rest[data]]]];
```

Since the list of parameters will be used multiple times, we will assign the list a name.

```
parameters = {a, b, c, d};
```

The following is the likelihood for the predicted value model, given the observed value y for a Poisson distribution. This expression will be used to construct a loglikelihood function.

```
pdffun[model_, y_] = PDF[PoissonDistribution[model], y]
```

$$\frac{e^{-\mathrm{model}}\mathrm{model}^{y}}{y!}$$

The elements of data are of the form {hmo, white, died, los}.

```
data[[1]]
```

{1., 1., 1., 10.}

The model of interest is los=$e^{a+b\mathrm{hmo}+c\mathrm{white}+d\mathrm{died}}$. The likelihood for the first element of data follows. Here an anonymous function (or pure function) is used to plug the first, second, and third elements of data[[1]] into the model, and the fourth element into the y argument of pdffun.

```
pdffun[Exp[a + b * #[[1]] + c * #[[2]] + d * #[[3]]], #[[4]]]&[data[[1]]]
```

$$2.75573 \times 10^{-7} e^{-e^{a+1.b+1.c+1.d}} \left(e^{a+1.b+1.c+1.d}\right)^{10.}$$

To compute all of the individual likelihoods, this pure function can be mapped onto data.

```
likelihoodlist = Map[pdffun[Exp[a + b * #[[1]] + c * #[[2]] + d * #[[3]]], #[[4]]]&, data];
```

The loglikelihood function is the sum of the logs of the individual likelihoods, and can be constructed as follows.

```
loglike = Total[Log[likelihoodlist]];
```

Good initial estimates for the values of a, b, c, and d can be obtained by performing a nonlinear least squares fitting of the model. The linear combination inside the exponential is constructed in the following manner by dotting the parameter vector with the vector of basic functions.

Figure 4(a). Poisson fitting.

lssquares = FindFit[data, Exp[parameters.Join[{1}, vars]], parameters, vars]

$\{a \to 3.31127, b \to -0.53037, c \to -0.685301, d \to -0.191712\}$

The loglikelihood can now be maximized using a Newton iteration with the FindMaximum function. The least squares estimates are taken as the starting values for the optimization. The output is the maximized loglikelihood and a list of rules for the parameter estimates.

{llval, parmvals} = FindMaximum[loglike, Transpose[{parameters, parameters/.lssquares}], Method → Newton]

$\{-1779.57, \{a \to 3.35621, b \to -0.500832, c \to -0.711893, d \to -0.272188\}\}$

One way to obtain the standard errors is to compute them from the information matrix. Here the information matrix is computed directly from the partial derivatives of the loglikelihood function.

infomatrix = −Table[*D*[loglike, parameters[[*i*]], parameters[[*j*]]], {*i*, 4}, {*j*, 4}]/.parmvals

{{3140., 311., 2885., 967.}, {311., 311., 293.619, 97.4504},
{2885., 293.619, 2885., 901.462}, {967., 97.4504, 901.462, 967.}}

The matrix can be displayed as a formatted matrix using the MatrixForm function.

MatrixForm[infomatrix]

$$\begin{pmatrix} 3140. & 311. & 2885. & 967. \\ 311. & 311. & 293.619 & 97.4504 \\ 2885. & 293.619 & 2885. & 901.462 \\ 967. & 97.4504 & 901.462 & 967. \end{pmatrix}$$

The standard errors are the square roots of the diagonal elements of the inverse of the information matrix. The following computes those values (note that for a matrix, Tr[matrix, List] extracts the diagonal elements).

sterrors = Sqrt[Tr[Inverse[infomatrix], List]]

{0.0635354, 0.0597689, 0.0653973, 0.0386774}

The likelihood ratio statistic is 2 times the difference between the loglikelihood for the fitted model and the loglikelihood for the model where each y is predicted by the mean. Here is that value.

lrstat = 2(llval − Total[Log[Map[pdffun[Mean[data[[All, −1]]], #]&, data[[All, −1]]]]])

239.775

The statistic in this example follows a chi-square distribution with 3 degrees of freedom. The p-value can be obtained using the CDF function.

1 − CDF[ChiSquareDistribution[3], lrstat]

0.

The parameters are asymptotically normally distributed, so confidence intervals can be constructed using the NormalCI function. The first argument of NormalCI is the parameter estimate, and the second argument is it's standard error estimate.

MapThread[NormalCI[#1, #2]&, {parameters/.parmvals, sterrors}]

{{3.23168, 3.48073}, {−0.617977, −0.383688}, {−0.840069, −0.583717}, {−0.347994, −0.196381}}

By default, a 95% confidence interval is returned, but other confidence levels can be specified via an option. These are the 99% confidence intervals.

MapThread[NormalCI[#1, #2, ConfidenceLevel → .99]&, {parameters/.parmvals, sterrors}]

{{3.19255, 3.51986}, {−0.654787, −0.346878}, {−0.880345, −0.543441}, {−0.371814, −0.172561}}

Figure 4(a). Continued. Poisson fitting.

Negative Binomial Fitting

The Poisson parameter estimates can be used as starting values for a negative binomial regression. The fitting process in this example will closely follow that of the Poisson fitting. First, we construct the likelihood function for a single observation. The following uses the common parameterization with an ancillary alpha parameter.

negbinompdffun[model_, y_] = PDF[NegativeBinomialDistribution[1/α, 1/(1 + α * model)], y]

$$\left(\frac{1}{1+\alpha \text{model}}\right)^{\frac{1}{\alpha}}\left(1-\frac{1}{1+\alpha \text{model}}\right)^{y} \text{Binomial}\left[-1+\frac{1}{\alpha}+y,-1+\frac{1}{\alpha}\right]$$

The loglikelihood function is constructed as before, except using the negative binomial likelihood rather than the Poisson likelihood. The loglikelihood function can be constructed in one line as follows. PowerExpand is used to convert terms of the form Log[Exp[x]] to x, a transformation that is not correct in general but is correct for loglikelihood expressions.

loglikenegbinom =
Total[PowerExpand[Log[Map[negbinompdffun[Exp[a + b * #[[1]] + c * #[[2]] + d * #[[3]]], #[[4]]]&, N[data]]]]];

The new list of parameters will include the ancillary alpha parameter.

nbparamters = {a, b, c, d, α};

The loglikelihood is now maximized using the values of a, b, c, and d from the Poisson fitting as the starting values a, b, c, and d in the negative binomial model. A small positive number can be used as a reasonable starting value for alpha.

{nbllval, nbparamvals} = FindMaximum[loglikenegbinom, Transpose[{nbparamters, {a, b, c, d, .1}/.lssquares}],
Method → Newton]

$\{-873.743, \{a \to 3.41915, b \to -0.482439, c \to -0.765011, d \to -0.317549, \alpha \to 0.655021\}\}$

The standard errors for a, b, c, d and alpha can be obtained from the information matrix for the negative binomial model. The following combines the information matrix computation, inversion, diagonal element extraction, and square root into one line of code.

nbsterrors =
Sqrt[Tr[Inverse[−Table[D[loglikenegbinom, nbparamters[[i]], nbparamters[[j]]], {i, 5}, {j, 5}]/.nbparamvals],
List]]

$\{0.270714, 0.153838, 0.270774, 0.113465, 0.0620183\}$

The 95% and 99% confidence intervals for a, b, c, and d can be computed in the same manner as in the Poisson example.

MapThread[NormalCI[#1, #2]&, {Most[nbparamters]/.nbparamvals, Most[nbsterrors]}]

$\{\{2.88856, 3.94974\}, \{-0.783956, -0.180921\}, \{-1.29572, -0.234304\}, \{-0.539937, -0.0951618\}\}$

MapThread[NormalCI[#1, #2, ConfidenceLevel → .99]&, {Most[nbparamters]/.nbparamvals, Most[nbsterrors]}]

$\{\{2.72184, 4.11646\}, \{-0.8787, -0.0861776\}, \{-1.46248, -0.0675442\}, \{-0.609816, -0.0252825\}\}$

Figure 4(b). Negative binomial fitting.

symbolic as well as numeric inputs. An additional feature of Mathematica's numeric functionality is the ability to use significance arithmetic. Mathematica allows for numbers of arbitrary precision in addition to machine numbers. Mathematica tracks the precision and accuracy of numeric computations not involving machine numbers, and returns as a result the correct precision and accuracy based on the intermediate computations.

Three additional capabilities that are both new to version 5.2 and that are of relevance to statisticians include:

a. Enhancements to various high-level functions, including series expansion of algebraic functions, derivatives, and singularity handling.

b. The incorporation of new algorithms for symbolic differential equations, which improve the ability to solve higher-order linear differential equations.

c. Enhanced quadratic quantifier elimination improves symbolic solving capabilities.

Some other items of note included in Mathematica are the Help Browser, Web services package, GUIKit, and the Notebook Indexer. The Help Browser is the help system in Mathematica. The contents of *The Mathematica Book*, documentation and examples for built-in functions, demos, documentation for packages, and advanced documentation are among the information included in the Help Browser. The Web services package provides an interface for obtaining XML data from the internet. GUIKit is a package for creating Java-based graphic user interfaces in Mathematica. The Notebook Indexer allows Notebook expressions to be searched by Google Desktop Search, Apple Spotlight, or Windows Desktop Search.

5. THE MATHEMATICA NOTEBOOK INTERFACE

A user interacts with the Mathematica computational kernel through the Notebook. Essentially, the Notebook is a fully integrated technical manuscript-preparation environment which manages everything one does in Mathematica. Formulas are entered and results displayed within the Notebook. Graphics are also both defined as well as displayed from within the Notebook. Additionally, the user may modify Notebook features in most any manner desired. Notebook images will be provided when I display example output in the following section.

I should mention here that Notebooks can be used as a complete document including text, computations, various cell styles, and typesetting. Users can design their own style sheets to create templates for their Notebooks. Notebooks can also be made into a slideshow, and converted to TEX, XML, or HTML. There is also an AuthorTools package included with Mathematica that can make the task of authoring in Mathematica a bit easier. The package contains a number of palettes allowing users to perform many formatting tasks by simply clicking on buttons.

6. EXAMPLES OF STATISTICAL CAPABILITY

Mathematica's numeric and symbolic capability is unparalleled as a desktop package. In this section I shall provide some examples of how Mathematica can be used to solve a few problems related to statistics. The examples will be annotated throughout, using Mathematica's Notebook interface.

I should mention here that these Notebook examples were saved in TEX format from within Mathematica, with graphics stored as .eps files. The TEX-exported Notebooks were then input directly into the LATEX document, which produced this review. The process was extremely easy to do.

The first example, shown in Figures 2(a)–(b), shows how to use Mathematica to perform some basic statistical calculations, including simple least squares regression. Figure 3 demonstrates how to generate random numbers from known distributions that are then modeled using Poisson and Beta maximum likelihood estimation, respectively. Poisson has one unknown parameter, mu; Beta has two unknown parameters, alpha and beta. Graphical representations are presented for both models.

The final example, shown in Figures 4(a)–(b), demonstrates how data stored in an Excel file can be read into Mathematica and modeled using Poisson and negative binomial regression. Having three binary predictors, initial values for the negative binomial model are calculated using Poisson regression, which are then used by the maximization function to calculate the maximum likelihood parameter estimates and standard errors for each of the predictors. The negative binomial ancillary parameter, called alpha, is also estimated. The values obtained in Mathematica agree with those obtained in Stata and LIMDEP.

7. SUMMARY REMARKS

Mathematica is not typically thought of as a statistical package. Most statisticians tend to use it as an ancillary package for symbolically solving derivatives and integrals, or for its matrix capabilities. However, I have shown that Mathematica can be used as a full-fledged statistical solution environment. Data may be imported into and exported from it in a wide variety of formats, manipulated in nearly any manner desired, and then may be subjected to a number of statistical analyses, including maximum likelihood estimation.

Aside from being able to access a huge variety of functions and distributions, Mathematica can easily be used to calculate mixtures of distributions, which can then be used to model data in ways unavailable—or which may involve substantial programming—in other commercial packages. As far as I am concerned, this makes Mathematica an extremely useful tool for professional statistician.

When I first thought of Mathematica for use as a complete statistical package, I assumed that only those with a solid background in mathematical statistics would be able to use its functionality. I love to design new statistical algorithms and procedures, but many statistical researchers do not. However, I found my fears to be mistaken. Those with a good undergraduate background in statistics can use Mathematica to perform nearly every statistical task they need for their research. Those having a more extensive mathematical background can use Mathematica to push the very limits of statistical theory and application.

REFERENCES

Beck, G. (managing editor), *The Mathematica Journal*, Champaign, IL: Wolfram Media. Available online at *www.mathematica-journal.com*.

Rose, C., and Smith, M. D. (2002), *Mathematical Statistics with Mathematica*, New York: Springer.

Wolfram, S. (2003), *The Mathematica Book* (5th ed.), Champaign, IL: Wolfram Media.