Using Open Source Software to Teach Mathematical Statistics

Douglas M. Bates

`bates@R-project.org`

University of Wisconsin – Madison

Outline

- Open Source Software for Statistics
  - What is **R**?
  - Obtaining and installing **R**
- An Example - MLE
- Conclusions from the example
- Links

Statistical software in Math Stat

- It is common to use some statistical software in applied courses to
  - work with realistic sized data sets
  - fit complicated models
  - gain insight from simulation
- Software in Math Stat courses has similar uses (especially simulation) but we must provide a simple interface for it to be useful.

Linking the computing to the text

It is important that the use of the computing system be integrated with the lectures and text. Two ways this can be accomplished are:

- Write a text that is tied to specific software and illustrates the use of that software, e.g. Nolan and Speed (2000)
- Adopt a conventional text but provide the examples from the text in the computing system.

Advantages of Open Source software

- No license fees or management of licenses.
  - We can install the software on all our departmental computers. We do not need to wait for a commerical software provider to port to a particular operating system or worry about their discontinuing a port.
  - It is much easier to convince other groups (Computer-Aided Engineering and the Computing Center, in my case) to install the software.
  - Students can install the software on their own computers without charge (and without violating licenses).
- Open Source projects encourage contributions from users so extensions are easier.

The S language

- It is a language and system developed by John Chambers and his co-workers at Bell Laboratories (formerly part of AT&T, now part of Lucent Technologies).
- The Association for Computing Machinery presented its 1999 Software System Award to John stating, “S has forever altered the way people analyze, visualize, and manipulate data . . . ”
- It is the *de facto* standard for computing in the Statistics research community.
- It is documented in many books, the best known of which are by Venables and Ripley (*Modern Applied Statistics with S-PLUS* and *S Programming*) and by Chambers (*Programming with Data*).

Design Goals for S

- To provide an environment for interactive computing with data.
- To allow users to transition easily into programmers as the need arises.
- To provide both exploratory and presentation graphics.
- To avoid reinventing existing tools.

Characteristics of S

- It is an interactive language based on functions and function calls.
- It provides an environment with persistent, self-describing objects. In particular, functions are first-class objects.
- It has a wide variety of graphics device drivers and controls.
- It provides interfaces to existing code and systems.

Implementations of S

- **S-PLUS**, sold by Insightful Corp. (formerly MathSoft, Inc.) and based on their exclusive license of the original **S** source from Lucent Technologies.
- **R**, an Open Source project conforming for the most part to the published descriptions of **S**.
  - Initially developed by Ross Ihaka and Robert Gentleman at the University of Auckland
  - Now developed and maintained by an widely-dispersed, international group of volunteers from academia and industry.
  - Operates through web sites (www.r-project.org), archives (cran.r-project.org) e-mail lists, CVS sites, rsync sites.

What is R?

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages
- A flexible data exchange mechanism accessing:

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages
- A flexible data exchange mechanism accessing:
  - text files and saved **R** workspaces

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages
- A flexible data exchange mechanism accessing:
  - text files and saved **R** workspaces
  - **S-PLUS** data objects, **SAS** XPORT datasets, **SPSS** saved datasets, Minitab worksheets, . . .

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages
- A flexible data exchange mechanism accessing:
  - text files and saved **R** workspaces
  - **S-PLUS** data objects, **SAS** XPORT datasets, **SPSS** saved datasets, Minitab worksheets, . . .
  - relational databases – ODBC, PostgreSQL, MySQL

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages
- A flexible data exchange mechanism accessing:
  - text files and saved **R** workspaces
  - **S-PLUS** data objects, **SAS** XPORT datasets, **SPSS** saved datasets, Minitab worksheets, . . .
  - relational databases – ODBC, PostgreSQL, MySQL
- An embeddable extension language

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages
- A flexible data exchange mechanism accessing:
  - text files and saved **R** workspaces
  - **S-PLUS** data objects, **SAS** XPORT datasets, **SPSS** saved datasets, Minitab worksheets, . . .
  - relational databases – ODBC, PostgreSQL, MySQL
- An embeddable extension language
- Part of the GNU (GNU's Not Unix) software system.

What is R?

- An Open Source implementation of John Chambers' award-winning **S** language
- A language and environment for data analysis and graphics
- A means of technology transfer through packages
- A flexible data exchange mechanism accessing:
  - text files and saved **R** workspaces
  - **S-PLUS** data objects, **SAS** XPORT datasets, **SPSS** saved datasets, Minitab worksheets, . . .
  - relational databases – ODBC, PostgreSQL, MySQL
- An embeddable extension language
- Part of the GNU (GNU's Not Unix) software system.
- Just install it and try it. Comes with a money-back guarantee.

How do I get R?

How do I get R?

- The informational web site `http://www.r-project.org/`

How do I get R?

- The informational web site `http://www.r-project.org/`
- **CRAN** - the Comprehensive R Archive Network

How do I get R?

- The informational web site `http://www.r-project.org/`
- **CRAN** - the Comprehensive R Archive Network
  - The primary site is `http://cran.r-project.org/`

How do I get R?

- The informational web site `http://www.r-project.org/`
- **CRAN** - the Comprehensive R Archive Network
  - The primary site is `http://cran.r-project.org/`
  - Mirror sites are available for many countries, e.g. `http://cran.us.r-project.org/`.

How do I get R?

- The informational web site `http://www.r-project.org/`
- **CRAN** - the Comprehensive R Archive Network
  - The primary site is `http://cran.r-project.org/`
  - Mirror sites are available for many countries, e.g. `http://cran.us.r-project.org/`.
- **CRAN** sites have binary distributions for Windows 95, 98, ME, NT4 and 2000 on Intel, for the Macintosh (System 8.6 to 9.1 and MacOS X), and for several Linux distributions.

How do I get R?

- The informational web site `http://www.r-project.org/`
- **CRAN** - the Comprehensive R Archive Network
  - The primary site is `http://cran.r-project.org/`
  - Mirror sites are available for many countries, e.g. `http://cran.us.r-project.org/`.
- **CRAN** sites have binary distributions for Windows 95, 98, ME, NT4 and 2000 on Intel, for the Macintosh (System 8.6 to 9.1 and MacOS X), and for several Linux distributions.
- New releases occur frequently - about every 3 months. Be prepared to re-install frequently.

Installing R

Installing R

Windows Download and run the installer, *SetupR.exe*.

Installing R

Windows Download and run the installer, *SetupR.exe*.

Windows Alternatively, see the *miniR* directory for floppy-sized images and their installer.

Installing R

Windows Download and run the installer, *SetupR.exe*.

Windows Alternatively, see the *miniR* directory for floppy-sized images and their installer.

Macintosh The distribution consists of one binhexed (.hxq) file that you expand using standard tools.

Installing R

Windows Download and run the installer, *SetupR.exe*.

Windows Alternatively, see the *miniR* directory for floppy-sized images and their installer.

Macintosh The distribution consists of one binhexed (.hxq) file that you expand using standard tools.

Linux RPM files are available for RedHat, SuSE, and Mandrake. Deb files are available for Debian. Under Debian you can list a **CRAN** archive in */etc/apt/sources.list* for automatic updates.

Installing R

Windows Download and run the installer, *SetupR.exe*.

Windows Alternatively, see the *miniR* directory for floppy-sized images and their installer.

Macintosh The distribution consists of one binhexed (.hxq) file that you expand using standard tools.

Linux RPM files are available for RedHat, SuSE, and Mandrake. Deb files are available for Debian. Under Debian you can list a **CRAN** archive in */etc/apt/sources.list* for automatic updates.

Unix Download and expand the compressed tar file of the sources. Run *./configure* then *make; make check; make install*

An Example - maximum likelihood

- Example is from a Mathematical Statistics course for Industrial Engineering students
- I use the text *Probability and Statistics for Engineering and the Sciences (5th ed)* by Jay Devore (Duxbury, 2000).
- Our I.E. Department requires that students learn maximum likelihood. Students are not at all enthusiastic.
- As is common, the text converts the problem of maximizing the likelihood or log-likelihood to the problem of solving the likelihood equations. Deriving the likelihood equations sometimes involves taking partial derivatives of complicated expressions. This is error-prone.

Weibull example from Devore (2000)

Let $X_1, \ldots, X_n$ be a random sample from a Weibull pdf

$$f(x;\alpha,\beta) = \begin{cases} \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Writing the likelihood and ln(likelihood), then setting both $(\partial/\partial\alpha)[\ln(f)] = 0$ and $(\partial/\partial\beta)[\ln(f)] = 0$ yields the equations

$$\alpha = \left[\frac{\sum x_i^\alpha \ln(x_i)}{\sum x_i^\alpha} - \frac{\sum \ln(x_i)}{n}\right]^{-1} \qquad \beta = \left(\frac{\sum x_i^\alpha}{n}\right)^{1/\alpha}$$

These two equations cannot be solved explicitly to give general formulas for the mle's $\widehat{\alpha}$ and $\widehat{\beta}$. Instead, for each sample $x_1, \ldots, x_n$, the equations must be solved using an iterative numerical procedure.

Weibull example in R

- Regard the problem of calculating MLEs as an optimization problem. Usually it is easier to optimize the log-likelihood rather than the likelihood itself. The parameter values that optimize the log-likelihood are also the MLEs.
- Probability density functions (probability functions for discrete distributions) have names starting with d — dnorm, dbinom, dunif, dweibull.
- All density functions take an optional argument log which, when TRUE causes evaluation of the log-density.
- The recipe becomes:

  > Regard the parameters as the variables and sum the log density for your distribution using your data.

An R session on the Weibull example

```
> library(Devore5)
> data(xmp04.30)    # import the data
> str(xmp04.30)     # examine the structure
‘data.frame’:   10 obs. of  1 variable:
 $ lifetime: num   282  501  741  851 1072 ...

> # Reasonable starting estimates are shape = 1, scale = 1000
> # Do a simple evaluation at this set of parameters
> sum(dweibull(xmp04.30$lifetime, shape=1, scale=1000, log=TRUE))
[1] -80.47655

> # Optimization functions minimize so use negative log-likelihood
> llfunc <- function(x) {   # express as a function
+  -sum(dweibull(xmp04.30$lifetime, shape=x[1], scale=x[2], log=TRUE))
+ }
> mle <- nlm(llfunc, c(shape = 1, scale = 1000), hessian = TRUE)
```

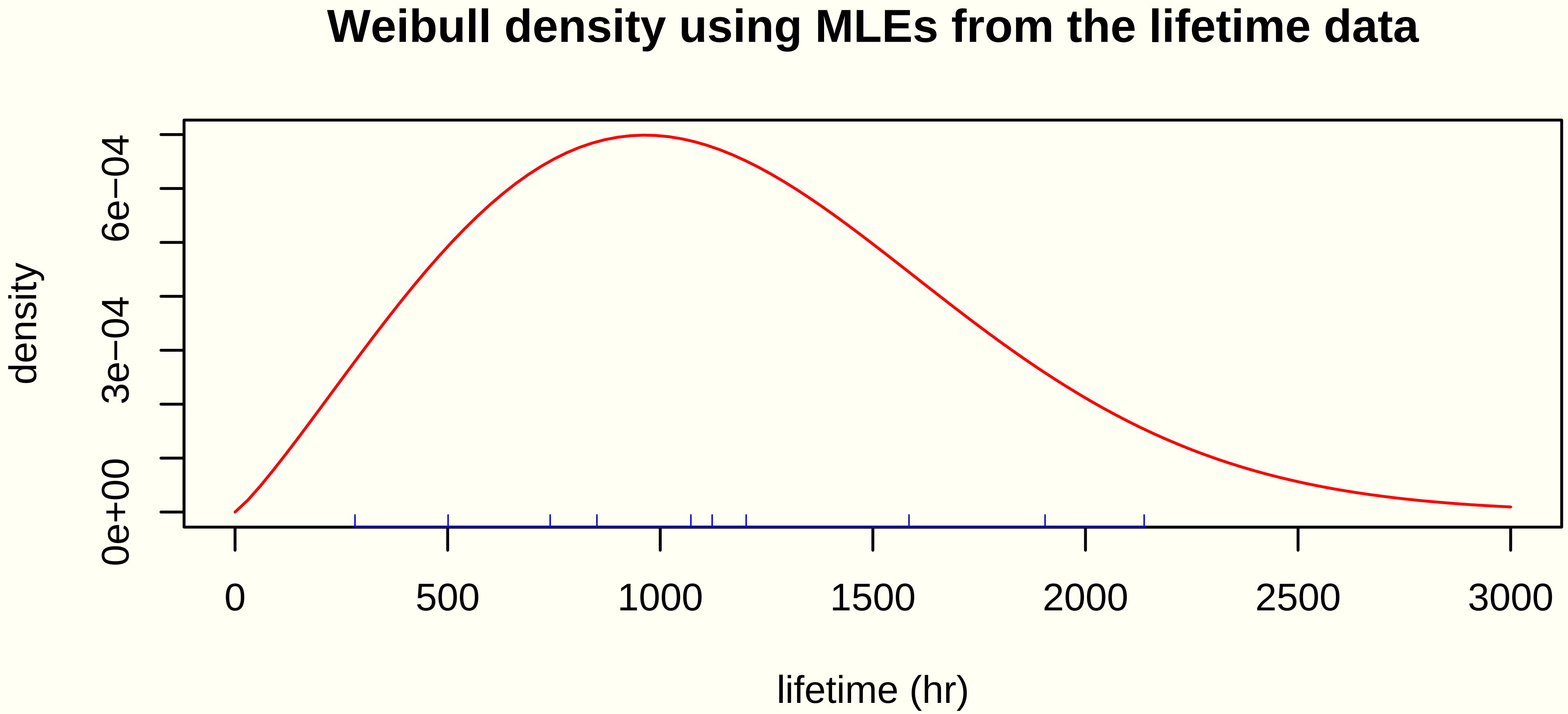
Results of the Weibull example

```
> str(mle)   # structure of the returned value
List of 6
 $ minimum   : num 77.1
 $ estimate  : num [1:2]    2.15 1289.34
 $ gradient  : num [1:2] -1.50e-05  2.98e-08
 $ hessian   : num [1:2, 1:2]  3.75e+00 -3.20e-03 -3.20e-03  2.78e-05
 $ code      : int 1
 $ iterations: int 20
> solve(mle$hessian)   # approximate variance-covariance matrix
           [,1]        [,2]
[1,]  0.2959642    34.06422
[2,] 34.0642200 39835.81998
```

We see that the maximum of the log-likelihood is -77.1, achieved at $\widehat{\alpha} = 2.15$ and $\widehat{\beta} = 1289.34$. The approximate standard errors of the estimates are $0.544 = \sqrt{0.29596}$ and $199.6 = \sqrt{39835.82}$. We can use the standard errors to determine a grid of (α, β) values for contouring the log-likelihood function.

Plotting the density at the estimates

```
> plot(function(x) dweibull(x, shape = 2.15, scale = 1289.34), 0, 3000
+       col = "red", xlab = "lifetime (hr)", ylab = "density",
+       main = "Weibull density using MLEs from the lifetime data")
> rug(xmp04.30$lifetime, col = "blue")
```

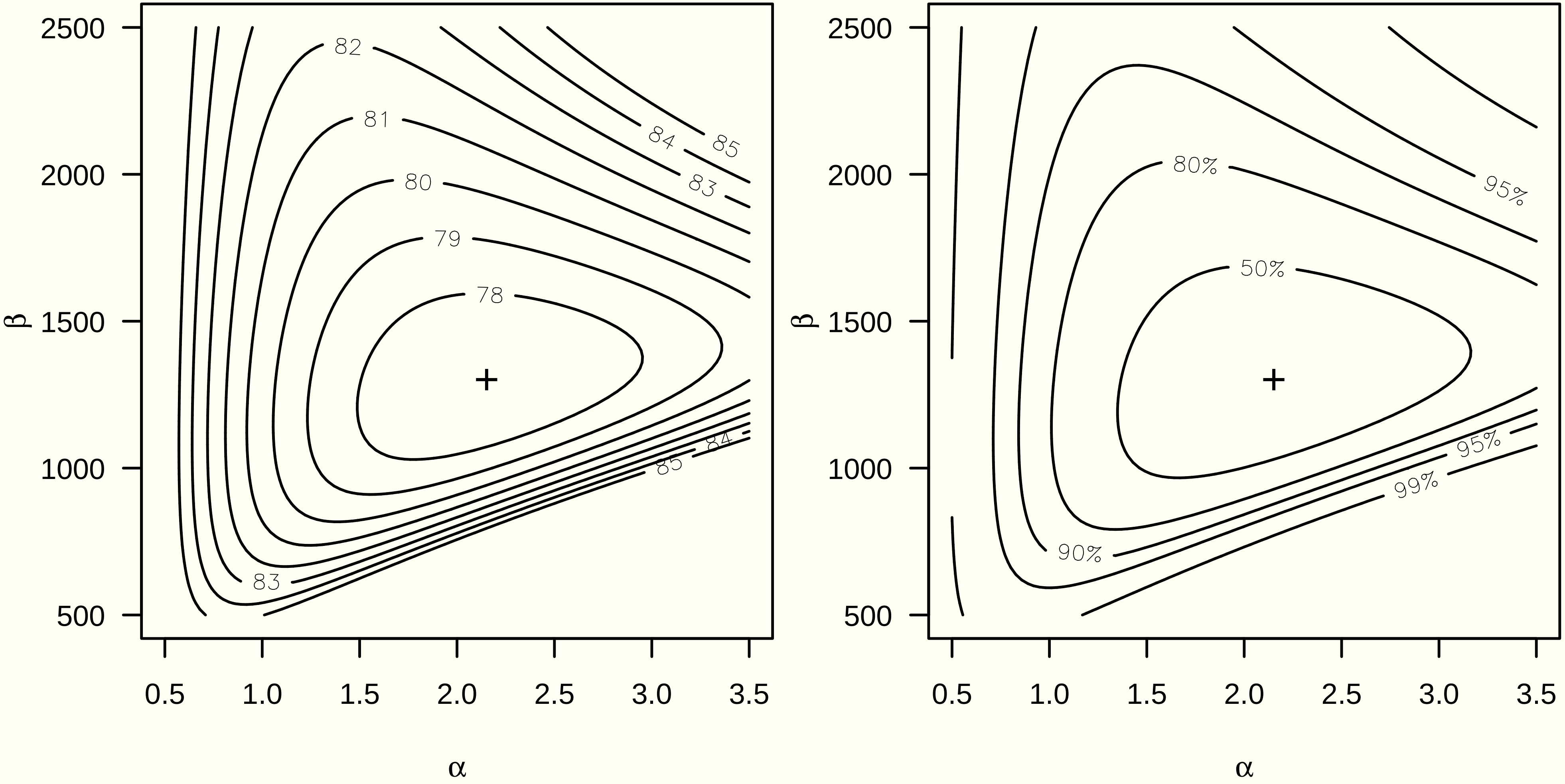


Contouring the log-likelihood function

```
> grid <- matrix(0.0, nrow = 101, ncol = 101)
> scvals <- seq(0.5, 3.5, len = 101)  # scale parameter
> shvals <- seq(500, 2500, len = 101) # shape parameter
> for (i in seq(along = scvals)) {
+    for (j in seq(along = shvals)) {
+        grid[i,j] <- llfunc(c(scvals[i], shvals[j]))
+    }
+ }
> contour(scvals, shvals, grid, levels = 77:85)
> points(mle$estimate[1], mle$estimate[2], pch = "+", cex = 1.5)
> title(xlab = expression(alpha), ylab = expression(beta))

> # Or use levels calculated from the chi-square distribution
> contour(scvals, shvals, grid,
+   levels = mle$min + qchisq(c(0.5,0.8,0.9,0.95,0.99), 2),
+   labels = paste(c(50,80,90,95,99), "%", sep = ""))
```

Log-likelihood contours - Weibull



Lessons from the Weibull example

- The likelihood function is the same as the probability density but with the parameters varying and the data fixed.
- For a random sample, the log-likelihood is

```
sum(d<distname>(<data>, par1, par2, ..., log = TRUE))
```

- We minimize the negative of the log-likelihood

```
llfunc <- function(x)
       -sum(d<distname>(<data>, par1 = x[1], ..., log = TRUE))
mle <- nlm(llfunc, <starting estimates>, hessian = TRUE)
```

- The inverse of the hessian provides an estimate of the variance-covariance matrix.
- For two-parameter models we can evaluate a grid of log-likelihood values and get contours.
- Standard errors from the inverse hessian are not always realistic indications of the variability in the parameter estimates.

A Method of Moments example

- Example 6.12 in Devore (2000) discussed method of moments estimates for the parameters in a γ distribution using some survival data.
- **R** provides facilities for a package author to document either functions or data sets. The example section of the documentation can be run in **R** using the example function.

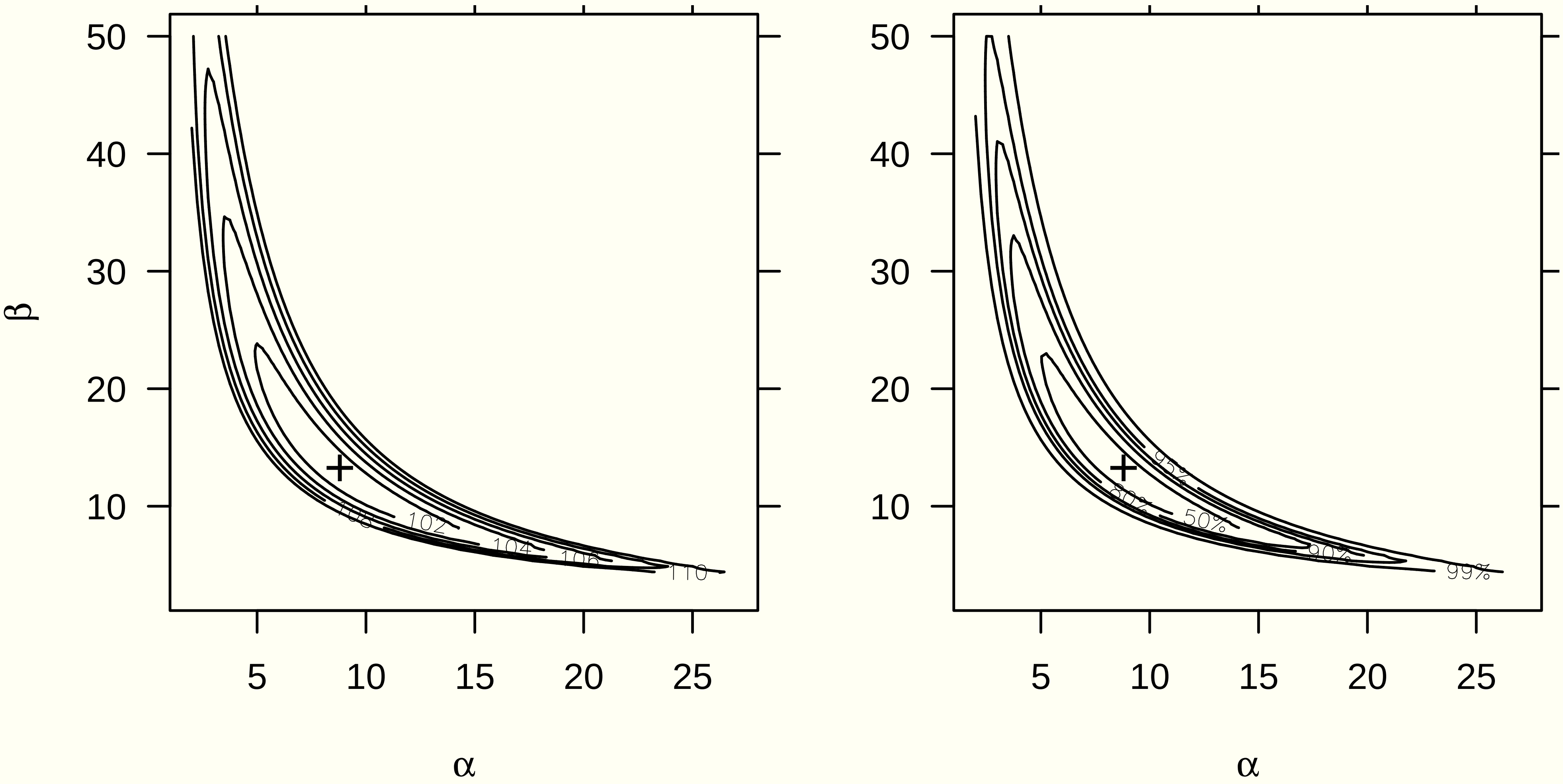
```
> example(xmp06.12)
x06.12> data(xmp06.12)
x06.12> gamma.MoM <- function(x) {
    xbar <- mean(x)
    mnSqDev <- mean((x - xbar)^2)
    c(alpha = xbar^2/mnSqDev, beta = mnSqDev/xbar)
}
x06.12> print(surv.MoM <- gamma.MoM(xmp06.12$Survival))
   alpha     beta
10.57725 10.72585
```

MLE for the gamma example

The same techniques used for the Weibull distribution can be applied to obtaining the MLEs for the parameters of the γ distribution

```
> llsurv <- function(x)
+  -sum(dgamma(xmp06.12$Survival, shape=x[1], scale=x[2], log = TRUE))
> mle2 <- nlm(llsurv, c(shape = 10, scale = 10), hessian = TRUE)
> mle2$estimate
[1]  8.799203 12.893209
> solve(mle2$hessian)
           [,1]      [,2]
[1,]   7.513112 -11.01259
[2,] -11.012585  17.08703
> shvals <- seq(2, 27, len = 101)
> scvals <- seq(3, 50, len = 101)
> for (i in seq(along = shvals)) {
+    for (j in seq(along = scvals)) {
+        grid[i,j] <- llsurv(c(shvals[i], scvals[j]))
+    }
+ }
> contour(shvals, scvals, grid, levels = seq(100, 110, 2))
```

Contours for the gamma log-likelihood



Conclusions from the gamma example

- Generalizing the Weibull example to other distributions (especially two-parameter distributions) is straightforward.
- The log-likelihood contours for this example are badly non-elliptical. Confidence intervals for α and β based on the standard error would have poor coverage.
- If you want to approximate the variability in the parameter estimates with symmetric intervals, you are better off using $\log\alpha$ and $\log\beta$.

If you want to see a really bad example, look at the log-likelihood contours for the parameters for the negative binomial using the data in Example 6.12 of Devore(2000).

Profile likelihood

In his description of the Weibull MLEs, Jay provides the expression

$$\beta = \left(\frac{\sum x_i^\alpha}{n}\right)^{1/\alpha}$$

showing that the MLE of β, conditional on α, can be calculated directly. This can be used to introduce profile likelihood

```
> profilell <- function(alpha)
+    -sum(dweibull(xmp04.30$lifetime, shape = alpha,
+         scale = mean(xmp04.30$lifetime^alpha)^(1/alpha), log = TRUE)
> profilell(2.15)   # negative profile log-likelihood at estimate
[1] 77.0951
> mle3 <- nlm(profilell, c(alpha = 1.0), hessian = TRUE)
> unlist(mle3[-3])
   minimum   estimate    hessian       code iterations
 77.095088   2.152001   3.378437   1.000000   7.000000
```

Profile likelihood in general

We can now discuss the use of profile likelihood to obtain confidence intervals with better coverage properties. For cases where there is no explicit expression for the conditional MLE, nested optimizations can be used.

This works in **R** but is very tricky, if not impossible, to do in **S-PLUS**.

What does software add to Math Stat?

- A rich set of distributions in our statistical software allows us to illustrate different distributions.
- Plotting and exploring densities, probability functions, cumulative distribution functions, empirical cdf's, etc. can help to internalize the concept of these functions as well as concepts like shape and scale parameters. (Recall how easy it was to plot the Weibull density.)
- We can treat MLEs as optimization problems and avoid the potentially confusing partial derivatives.
- We can illustrate examples for which there are no analytic solutions.
- We can discuss more advanced concepts (confidence regions based on contours of the log-likelihood, profile likelihood, likelihood-based confidence intervals that do not have to be symmetric) in elementary courses.

Sources of information about R

- The web site `http://www.r-project.org/` and **CRAN**
- The frequently asked questions (FAQ) list at `http://www.ci.tuwien.ac.at/~hornik/R/`, mirrored at `http://cran.r-project.org/doc/FAQ/`, is a great source of information, especially for users switching from **S-PLUS**.
- The manuals in the documentation directory `http://cran.r-project.org/doc/manuals`. See especially *R-intro.pdf* and *R-data.pdf* .