Maximum Likelihood Estimation

Greene Ch.14; App. E
R script `mod2s1a, mod2s1b`

If we feel safe making assumptions on the statistical distribution of the error term, Maximum Likelihood Estimation (MLE) is an attractive alternative to Least Squares for linear regression models. Even better, MLE can also be used for non-linear models. It is thus a more generally applicable estimation strategy than Ordinary Least Squares.

As with any econometric estimation, we start with a stipulated population model, including distribution assumptions for the error term. Sticking with the CLRM, we have

$$\begin{aligned} y_i &= \mathbf{x_i'}\boldsymbol{\beta} + \varepsilon_i \qquad \varepsilon_i \sim n\left(0,\sigma^2\right) \\ \mathbf{y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \qquad \boldsymbol{\varepsilon} \sim n\left(\mathbf{0},\sigma^2\mathbf{I}\right) \end{aligned} \tag{1}$$

As evident from the full-sample-model, we continue to assume that individual errors are identically and independently distributed (i.i.d) following a normal distribution with mean 0 and variance σ^2.

Let's collect the unknown parameters $\boldsymbol{\beta}$ and σ^2 into a single vector $\boldsymbol{\theta}$. Then, the density for a single observation (loosely translated as "the probability of observing a given observation") conditional on $\boldsymbol{\theta}$ (and, of course also on $\mathbf{x_i}$, which we will tacitly assume throughout) is given by

$$f\left(y_i \mid \boldsymbol{\theta}\right) = \left(2\pi\sigma^2\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\left(\frac{y_i - \mathbf{x}_i'\boldsymbol{\beta}}{\sigma}\right)^2\right) \tag{2}$$

Since the ε_i's are independent, so are the y_i's. We can thus write the density for the entire sample (the "sample density" or "sample distribution") as a product of individual densities, i.e.

$$\begin{aligned} &f\left(\mathbf{y} \mid \boldsymbol{\theta}\right) = \prod_{i=1}^{n} f\left(y_i \mid \boldsymbol{\theta}\right) = \left(2\pi\sigma^2\right)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}\left(y_i - \mathbf{x_i'}\boldsymbol{\beta}\right)^2\right) = \\ &\left(2\pi\sigma^2\right)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)\right) \end{aligned} \tag{3}$$

The last equality follows from the fact that exp(a)*exp(b)=exp(a+b), and using the inner product-of vector rule for the summation of squared products.

Our focus will lie on the estimation of the parameter vector $\boldsymbol{\theta}$. For that purpose, we can interpret $f\left(\mathbf{y} \mid \boldsymbol{\theta}\right)$ as a function of $\boldsymbol{\theta}$, given the data. This change of estimation focus is expressed via a change in notation. Specifically, we will write (2) as $l\left(\boldsymbol{\theta} \mid y_i\right) = l_i\left(\boldsymbol{\theta}\right)$, and refer to it as the *likelihood function for a* *single observation.* Analogously, we will write (3) as $L(\boldsymbol{\theta}|\mathbf{y})=L(\boldsymbol{\theta})$, and call it the *likelihood function for the entire sample*, or simply the *sample likelihood.*[1]

By convention and for mathematical convenience (e.g. to avoid very large numbers), we work with the likelihood in log form. My notation will be as follows:

$l_i(\boldsymbol{\theta})$ is called the "likelihood function" (LF) for a single observation. $L(\boldsymbol{\theta})$ is the sample LF.
$\ln l_i(\boldsymbol{\theta})$ is the log-likelihood function (LLF) for a single observation. $\ln L(\boldsymbol{\theta})$ is the LLF for the sample.

To derive the sample LLF, you can take one of two routes: (i) take the log of the LF for an individual observations, then add over individuals (since ln(a*b) = ln(a) + ln(b)), or compute the sample LF first, then take the log. For the normal regression model:

$$\begin{aligned}\ln l_i(\boldsymbol{\theta})&=-\tfrac{1}{2}\ln 2\pi-\tfrac{1}{2}\ln(\sigma^2)-\frac{1}{2}\left(\frac{y_i-\mathbf{x}_i'\boldsymbol{\beta}}{\sigma}\right)^2\\ \ln L(\boldsymbol{\theta})&=-\tfrac{n}{2}\ln 2\pi-\tfrac{n}{2}\ln(\sigma^2)-\frac{1}{2\sigma^2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})\end{aligned} \tag{4}$$

Once the LLF is specified, the goal is to find the $\boldsymbol{\theta}$ that maximizes this LLF for the sample. Thus, we treat the sample LLF as an objective function, which we are trying to maximize. Note: Most individual density ordinates comprised in $L(\boldsymbol{\theta})$ will be much smaller than 1, so $L(\boldsymbol{\theta})$ itself, being a product of these ordinates *usually* takes a value between 0 and 1. The log of this will thus be negative. Therefore, maximizing $\ln L(\boldsymbol{\theta})$ means finding a value of $\boldsymbol{\theta}$ that gets the LLF "as close to zero as possible" from below.

The first derivative of $\ln l_i(\boldsymbol{\theta})$ is the individual level score function or "gradient", denoted as $g_i(\boldsymbol{\theta})$. The sample equivalent is given as $g(\boldsymbol{\theta})=\sum_{i=1}^{n}g_i(\boldsymbol{\theta})$. These gradient expressions always have the same dimension as $\boldsymbol{\theta}$. Again, you have two choices for computing $g(\boldsymbol{\theta})$: (i) Compute the individual gradient, then add up over observations, or compute the sample gradient directly from $\ln L(\boldsymbol{\theta})$. The first order conditions for optimization require the sample gradient to equal zero at the solution values for $\boldsymbol{\theta}$. For the normal regression model:

$$\begin{bmatrix}\dfrac{\partial \ln L(\boldsymbol{\theta})}{\partial \boldsymbol{\beta}}\\ \dfrac{\partial \ln L(\boldsymbol{\theta})}{\partial \sigma^2}\end{bmatrix}=\begin{bmatrix}\dfrac{\mathbf{X}'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^2}\\ -\dfrac{n}{2\sigma^2}+\dfrac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{2\sigma^4}\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\0\end{bmatrix} \tag{5}$$

[1] Strictly speaking, the likelihood function is only *proportional* to the sample distribution. This means it only includes elements of the sample distribution that contain the parameters of interest. Any parts of the sample distribution that can be multiplicatively separated from those elements are ignored in the likelihood function. However, whichever estimate of $\boldsymbol{\theta}$ that maximizes the sample distribution will also maximize the likelihood function, and vice versa. Thus, this distinction can be neglected for practical purposes. Here I will follow our main textbook and use the same mathematical expression for sample distribution and likelihood.

The first order conditions are also called "*Likelihood Equations*". They lead to the maximum likelihood estimators

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X'X}\right)^{-1}\mathbf{X'y} \quad \text{and} \quad \hat{\sigma}^2 = \frac{\mathbf{e'e}}{n} \quad \textit{where} \quad \mathbf{e} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}} \tag{6}$$

Clearly, the solution for the coefficient vector is identical to the one derived from the OLS problem. The estimate for σ^2 differs slightly from the OLS solution as it does not correct the denominator for degrees of freedom (*k*).

To assure a maximum, we need to examine the properties of the Hessian matrix of second derivatives. We could again derive the this expression for a single observation (denoted $H_i\left(\boldsymbol{\theta}\right)$), then add up over all observations, or compute the sample Hessian $H\left(\boldsymbol{\theta}\right)$ directly from the sample gradient. For the normal regression model:

$$H\left(\boldsymbol{\theta}\right) = \begin{bmatrix} \dfrac{\partial \ln L\left(\boldsymbol{\theta}\right)}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}'} & \dfrac{\partial \ln L\left(\boldsymbol{\theta}\right)}{\partial\boldsymbol{\beta}\partial\sigma^2} \\ \dfrac{\partial \ln L\left(\boldsymbol{\theta}\right)}{\partial\sigma^2\partial\boldsymbol{\beta}'} & \dfrac{\partial \ln L\left(\boldsymbol{\theta}\right)}{\partial\left(\sigma^2\right)^2} \end{bmatrix} = \begin{bmatrix} -\dfrac{\mathbf{X}'\mathbf{X}}{\sigma^2} & -\dfrac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sigma^4} \\ -\dfrac{\boldsymbol{\varepsilon}'\mathbf{X}}{\sigma^4} & -\dfrac{2\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} - \sigma^2 n}{2\sigma^6} \end{bmatrix} \text{where} \quad \boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta} \tag{7}$$

It can be shown that all eigenvalues for this Hessian are negative, thus $H\left(\boldsymbol{\theta}\right)$ is negative definite, and we have indeed a maximum.

Regularity and Related Properties of ML Estimation

For a function $\prod_{i=1}^{n} f\left(y_i \mid \boldsymbol{\theta}\right)$ to be amenable to ML estimation, it must satisfy the following "*regularity conditions*" (see Greene p. 515, although he expresses these conditions differently)

1. The first three derivatives of $\ln f\left(y_i \mid \boldsymbol{\theta}\right)$ w.r.t. $\boldsymbol{\theta}$ are continuous and finite for almost all y_i and for all elements of $\boldsymbol{\theta}$, and the derivatives are integrable (so we can derive the expectations of the first and second derivatives - see below)

2. The support of y_i does not depend on $\boldsymbol{\theta}$. (So no element in $\boldsymbol{\theta}$ can denote a bound of the distribution of y_i)

3. The true value of $\boldsymbol{\theta}$ lies in a closed and bounded "compact set". (i.e. there can't be discontinuous "sets" of candidates for the solution of $\boldsymbol{\theta}$)

If these regularity conditions are satisfied, two interesting properties of ML estimation arise. These are the *score identity* and the *information matrix identity*.

Score identity

The score identity states that the expectation of the gradient (w.r.t. y_i) at the *true values of the parameters* is zero. This holds for both an individual gradient and the sample gradient. In mathematical terms:

$$E_{y_i}\left(g_i\left(\boldsymbol{\theta}\right)\right)=\mathbf{0} \quad E_{\mathbf{y}}\left(g\left(\boldsymbol{\theta}\right)\right)=\mathbf{0} \tag{8}$$

Let's verify this for the normal regression model:

$$E_{\mathbf{y}}\begin{bmatrix} \dfrac{\mathbf{X}'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)}{\sigma^2} \\ -\dfrac{n}{2\sigma^2}+\dfrac{\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)}{2\sigma^4}\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\0\end{bmatrix} \tag{9}$$

First term:

$$E\left(\frac{\mathbf{X}'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)}{\sigma^2}\right)=\frac{1}{\sigma^2}\mathbf{X}'E\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right) \qquad \text{and}$$

$$E\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)=E\left(\mathbf{y}\right)-\mathbf{X}\boldsymbol{\beta}=\mathbf{X}\boldsymbol{\beta}-\mathbf{X}\boldsymbol{\beta}=\mathbf{0}$$

Second term:

$$E\left(-\frac{n}{2\sigma^2}+\frac{\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)}{2\sigma^4}\right)=-\frac{n}{2\sigma^2}+\frac{1}{2\sigma^4}E\left(\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)\right) \quad \text{and}$$

$$\mathrm{E}\left(\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)\right)=E\left(\mathbf{y}'\mathbf{y}-2\mathbf{y}'\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}\right)=E\left(\mathbf{y}'\mathbf{y}\right)-2E\left(\mathbf{y}'\right)\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}=$$

$$E\left(\mathbf{y}'\mathbf{y}\right)-\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \qquad \textit{where}$$

$$E\left(\mathbf{y}'\mathbf{y}\right)=E\left(\left(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}\right)'\left(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}\right)\right)=E\left(\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}+2\boldsymbol{\beta}'\mathbf{X}'\boldsymbol{\varepsilon}+\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon}\right)=\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}+n\sigma^2$$

$$\text{So:} \qquad \mathrm{E}\left(\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)\right)=\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}-\boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}+n\sigma^2=n\sigma^2 \quad \text{and}$$

$$E\left(-\frac{n}{2\sigma^2}+\frac{\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)'\left(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}\right)}{2\sigma^4}\right)=-\frac{n}{2\sigma^2}+\frac{n}{2\sigma^2}=0$$

Information Matrix Identity

The *Information Matrix* is the *negative* of the *expectation* of the Hessian. At the individual level, we will denote it as $I_i\left(\boldsymbol{\theta}\right)$. If we sum this over i, we obtain the sample Information matrix $I\left(\boldsymbol{\theta}\right)$. Alternatively, we can derive $I\left(\boldsymbol{\theta}\right)$ by directly taking the expectation (w.r.t. **y**) of the sample Hessian:

$$I(\boldsymbol{\theta})=-E(H(\boldsymbol{\theta}))=-E_{\mathbf{y}}\begin{bmatrix}-\frac{\mathbf{X}'\mathbf{X}}{\sigma^2} & -\frac{\mathbf{X}'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})}{\sigma^4}\\ -\frac{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'\mathbf{X}}{\sigma^4} & -\frac{2(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})'(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})-\sigma^2 n}{2\sigma^6}\end{bmatrix}=$$
$$\begin{bmatrix}\frac{\mathbf{X}'\mathbf{X}}{\sigma^2} & 0\\ 0 & \frac{2n\sigma^2-\sigma^2 n}{2\sigma^6}\end{bmatrix}=\begin{bmatrix}\frac{\mathbf{X}'\mathbf{X}}{\sigma^2} & 0\\ 0 & \frac{n}{2\sigma^4}\end{bmatrix} \tag{10}$$

As we will learn shortly, the inverse of $I(\boldsymbol{\theta})$ is the most efficient estimator for the variance-covariance matrix of $\boldsymbol{\theta}$.

The Information Matrix Identity states that at the *true parameter values*, the variance of the gradient is equal to the information matrix, i.e.

$$I_i(\boldsymbol{\theta}\,|\,y_i)=Var\left[g_i(\boldsymbol{\theta})\right]=E_{y_i}\left[g_i(\boldsymbol{\theta})\cdot g_i(\boldsymbol{\theta})'\right] \quad \text{and}$$
$$I(\boldsymbol{\theta}\,|\,\mathbf{y})=Var\left[g(\boldsymbol{\theta})\right]=E_{\mathbf{y}}\left[g(\boldsymbol{\theta})\cdot g(\boldsymbol{\theta})'\right] \tag{11}$$

This is a bit tedious to show for the normal regression model, but we will illustrate this equality in other examples.

Computational Implementation of MLE

For most econometric problems analytical solutions for $\hat{\boldsymbol{\beta}}$ and associated statistics are difficult to derive. Instead, we let the computer solve the problem using an "iterative algorithm". The basic concept is as follows:

1. Define some (*k*x1) starting vector $\boldsymbol{\theta}_\mathbf{0}$ (e.g. using OLS, or results form a different data set, or form a related but simpler model, etc). Also, choose a "*convergence criterion*" that determines when the algorithm is completed (ex: stop if the change in $\ln L(\boldsymbol{\theta}\,|\,\mathbf{y})$ by moving from one candidate $\boldsymbol{\theta}$ to another is smaller than "*c*", where *c* is some small number, usually between 0.0001 and 0.01.

2. Move from $\boldsymbol{\theta}_\mathbf{0}$ to $\boldsymbol{\theta}_\mathbf{1}$ (and, more generally, from $\boldsymbol{\theta}_\mathbf{t}$ to $\boldsymbol{\theta}_{\mathbf{t+1}}$) using the following rule:
$$\boldsymbol{\theta}_{\mathbf{t+1}}=\boldsymbol{\theta}_\mathbf{t}+\lambda_t\boldsymbol{\Delta}_\mathbf{t} \tag{12}$$
where $\boldsymbol{\Delta}_\mathbf{t}$ is a k x k matrix of k *direction vectors* (one for each element of $\boldsymbol{\theta}$), and scalar λ_t is the "step size" that determines how far along the directionals we move until we determine that we have reached $\boldsymbol{\theta}_\mathbf{t}$.

3. Evaluate $\ln L(\boldsymbol{\theta}_\mathbf{t}\,|\,\mathbf{y})$ and $\ln L(\boldsymbol{\theta}_{\mathbf{t+1}}\,|\,\mathbf{y})$ and determine if the difference ($\ln L(\boldsymbol{\theta}_\mathbf{t}\,|\,\mathbf{y})$ - $\ln L(\boldsymbol{\theta}_{\mathbf{t+1}}\,|\,\mathbf{y})$)< 0 (in which case the move from $\boldsymbol{\theta}_\mathbf{t}$ to $\boldsymbol{\theta}_{\mathbf{t+1}}$ has brought us to a higher point on *ln L*) or not. If not,

repeat steps 2) and 3) until the difference < 0. Together, Steps 2) and 3) form a single "*iteration*".

4. Continue until $\ln L(\boldsymbol{\theta}_{\mathbf{t+1}} \mid \mathbf{y}) - \ln L(\boldsymbol{\theta}_{\mathbf{t}} \mid \mathbf{y}) < c$ (or some other convergence criterion is satisfied). The last candidate $\boldsymbol{\theta}$ is your MLE solution.

Now we need to address the choice of λ_t and $\boldsymbol{\Delta}_{\mathbf{t}}$. As described in more detail in Greene's Appendix E, a popular direction matrix that works well in many practical applications is

$$\boldsymbol{\Delta}_{\mathbf{t}} = \left(-H(\boldsymbol{\theta}_{\mathbf{t}})\right)^{-1} g(\boldsymbol{\theta}_{\mathbf{t}}) \qquad (13)$$

This is called "*Newton's Method*". The line search parameter λ_t is either chosen ex ante (i.e. λ_t =0.5), or determined at each iteration by satisfying

$$\frac{\partial \ln L(\boldsymbol{\theta}_{\mathbf{t}} + \lambda_t \boldsymbol{\Delta}_{\mathbf{t}})}{\partial \lambda_t} = 0 \qquad (14)$$

As evident from (13) Newton's approach requires the evaluation of the gradient and Hessian at each iteration. If the analytical forms for *g* and *H* can be easily derived, or if good numerical approximations are available, this is not a problem. However, in some cases *H* may be difficult to compute or even approximate. In such situations the $\left(-H(\boldsymbol{\theta}_{\mathbf{t}})\right)^{-1}$ term in (13) can be replaced by the "outer product of gradients" (OPG), given as

$$OPG(\boldsymbol{\theta}_{\mathbf{t}}) = \left(G(\boldsymbol{\theta}_{\mathbf{t}})' G(\boldsymbol{\theta}_{\mathbf{t}})\right)^{-1} \text{ where } \underset{nxk}{G(\boldsymbol{\theta}_{\mathbf{t}})} = \begin{bmatrix} g_1(\boldsymbol{\theta}_{\mathbf{t}})' \\ g_2(\boldsymbol{\theta}_{\mathbf{t}})' \\ \vdots \\ g_n(\boldsymbol{\theta}_{\mathbf{t}})' \end{bmatrix} \qquad (15)$$

This procedure is not quite as accurate as using the actual Hessian, but it works sufficiently well in many applications. We will re-visit the OPG shortly when we talk about estimation of the asymptotic variance of $\hat{\boldsymbol{\beta}}$.

Estimating the Asymptotic Variance of the ML Estimator

To recap, for the OLS estimator **b** we were able to derive the exact ("finite sample") variance as $V(\mathbf{b}) = \sigma^2 (\mathbf{X}'\mathbf{X})^{-1}$, which we then approximated by $s^2 (\mathbf{X}'\mathbf{X})^{-1}$. However, s^2 was shown to be unbiased, which is another finite sample property. In other words, we never had to resort to large-sample (or "asymptotic") theory to derive V(**b**).

This is different for the variance of the ML estimator, $V(\hat{\boldsymbol{\beta}})$. For this construct, no finite sample results are available. All estimators for $V(\hat{\boldsymbol{\beta}})$ are asymptotic in nature, i.e. they converge to the true value as the sample size goes to infinity. Thus, they become more reliable with larger sample size.

The most commonly used estimator for $V\left(\hat{\boldsymbol{\beta}}\right)$ is the inverse of the negative Hessian at the solution value, i.e.

$$\hat{V}\left(\hat{\boldsymbol{\beta}}\right)=\left(-H\left(\hat{\boldsymbol{\beta}}\right)\right)^{-1} \tag{16}$$

A second estimator that does not require the computation of the Hessian is the inverted OPG at the solution value, i.e.

$$\hat{\hat{V}}\left(\hat{\boldsymbol{\beta}}\right)=\left(\mathbf{G}'\mathbf{G}\right)^{-1} \tag{17}$$

This approach is also known as the "BHHH" estimator, named after the founding authors in Berndt. et al. (1974). It is certainly convenient, but can be very inaccurate for smaller samples. For a related discussion see Greene p. 522.

Notes for R Implementation

Scripts `mod2s1a` and `1b` estimate the same CLRM based on wage data as script `mod1s2b` via MLE. As we will learn shortly, if the CLRM assumptions are satisfied, OLS and MLE should produce basically identical results under large sample sizes. This is indeed the case for this example.

Script `mod2s1a` calls **R**'s built-in optimization routine ("`optim`") , which uses a numerical gradient and Hessian to solve the optimization problem. This is convenient in for the (many) instances when the analytical gradient and / or Hessian are difficult to compute or program.

You can also run `optim` with a user-supplied gradient (and let just the Hessian be derived numerically). See the `optim` – manual (type "?optim" in R) for details.

As a general rule, the more analytical components you can supply, the faster and more accurate your algorithm will be.

A fully analytical implementation of MLE is given in script `mod2s1b,` where we code up our own Newton routine based on analytical gradient and Hessian to find the maximum of the log-likelihood function.