Part I

PRELIMINARIES

COPYRIGHTED MATERIAL

1

A taste of likelihood

When it is not in our power to follow what is true, we ought to follow what is most probable. – René Descartes

1.1 Introduction

The word *likelihood* has its origins in the late fourteenth century (Simpson and Weiner 1989), and examples of its usage include as an indication of probability or promise, or grounds for probable inference. In the early twentieth century, Sir Ronald Fisher (1890–1962) presented the ‘absolute criterion’ for parameter estimation (Fisher 1912), and some nine years later he gave this criterion the name *likelihood* (Fisher 1921, Aldrich 1997). Fisher’s choice of terminology was ideal, because the centuries-old interpretation of the word *likelihood* is also applicable to the formal statistical definition of likelihood that is used throughout this book.

Here, likelihood is used within the traditional framework of frequentist statistics, and maximum likelihood (ML) is presented as a general-purpose tool for inference, including the evaluation of statistical significance, calculation of confidence intervals (CIs), model assessment, and prediction. The frequentist theory underpinning the use of maximum likelihood is covered in Part III, where it is seen that maximum likelihood estimators (MLEs) have optimal properties for sufficiently large sample sizes. It is for this reason that maximum likelihood is the most widely used form of traditional parametric inference. The pragmatic use of ML inference is the primary focus of this book and is covered in Part II. The reader who is already comfortable with the concept of likelihood and its basic properties can proceed to Part II directly.

Maximum Likelihood Estimation and Inference: With Examples in R, SAS and ADMB, First Edition. Russell B. Millar.
© 2011 John Wiley & Sons, Ltd. Published 2011 by John Wiley & Sons, Ltd.

Likelihood is also a fundamental concept underlying other statistical paradigms, especially the Bayesian approach. Bayesian inference is not considered here, but consideration of the philosophical distinctions between frequentist and Bayesian statistics is examined in Chapter 14. In addition, it is seen that some maximum likelihood methodology can be motivated using Bayesian considerations. This includes techniques for prediction (Section 4.6), and the use of integrated likelihood (Section 9.3).

A simple binomial example (Example 1.1) is used in Section 1.2 to motivate and demonstrate many of the essential properties of likelihood that are developed in later chapters. In this example, the likelihood is simply the probability of observing $y = 10$ successes from 100 trials. The fundamental conceptual point is that likelihood expresses the probability of observing 10 successes as a function of the unknown success probability p. That is, the likelihood function does not consider other values of y. It takes the knowledge that $y = 10$ was the observed number of successes and it uses the binomial probability of the outcome $y = 10$, evaluated at different possible values of p, to judge the relative likelihood of those different values of p.

1.2 Motivating example

Throughout this book, adding a zero subscript to a parameter (e.g. p_0) is used generically to denote a specified value of the parameter. This is typically either its true unknown value, or a hypothesized value.

1.2.1 ML estimation and inference for the binomial

Example 1.1 applies ML methodology to the binomial model in order to obtain the MLE of the binomial probability, the standard error of the MLE, and confidence intervals. This example is revisited and extended in subsequent chapters. For example, Sections 4.2.2 and 4.3.1 look at issues concerning approximate normality of the MLE, and Example 4.10 considers prediction of a new observation from the binomial distribution.

Example 1.1. Binomial. A random sample of one hundred trials was performed and ten resulted in success. What can be inferred about the unknown probability of success, p_0?

For any potential value of p $(0 \leq p \leq 1)$ for the probability of success, the probability of y successes from n trials is given by the binomial probability formula (Section 15.4.1). With $y = 10$ successes from $n = 100$ trials, this is

$$\begin{aligned} L(p) &= \text{Prob}(10 \text{ successes}) \\ &= \frac{100!}{90!\,10!}\, p^{10}(1-p)^{90} \\ &= 1.731 \times 10^{13} \times\ p^{10}(1-p)^{90}\,, \quad 0 \leq p \leq 1\,. \end{aligned} \tag{1.1}$$

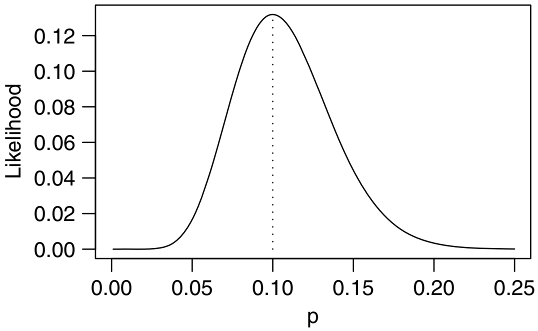


Figure 1.1 Binomial likelihood for 10 successes from 100 trials.

The above probability is the likelihood, and has been denoted $L(p)$ to make its dependence on p explicit.

A plot of $L(p)$ (Figure 1.1) shows it to be unimodal with a peak at $p = 0.1$. This is the MLE and will be denoted $\widehat{p}$. For the binomial model, the MLE of the probability of success is always the observed proportion of successes $\widehat{p} = y/n$ (Example 2.5). □

> The curve in Figure 1.1 looks somewhat like the bell-shaped curve of the normal density function. However, it is not a density (it is a likelihood function) and nor is it bell-shaped. On close inspection it can be seen that the curve is slightly right-skewed.

Box 1.1

In the above example, the MLE $\widehat{p}$ is simply a point-estimate of p_0, and is of limited use without any sense of how reliable it is. For example, it would be more meaningful to have a range of plausible values of the unknown p_0, or to know if some pre-specified value, e.g. $p_0 = 0.5$, was reasonable. Such questions can be addressed by examining the shape of the likelihood function, or more usually, the shape of the log-likelihood function.

The (natural) log of the likelihood function is used far more predominantly in likelihood inference than the likelihood function itself, for several good reasons:

1. The likelihood and log-likelihood are both maximized by the MLE.
2. Likelihood values are often extremely small (but can also be extremely large) depending on the model and amount of data. This can make numerical optimization of the likelihood highly problematic, compared to optimization of the log-likelihood.

3. The plausibility of parameter values is quantified by ratios of likelihood (Section 2.3), corresponding to a difference on the log scale.

4. It is from the log-likelihood (and its derivatives) that most of the theoretical properties of MLEs are obtained (see Part III).

The theoretical properties alluded to in Point 4 are the basis for the two most commonly used forms of likelihood inference – inference based on the likelihood ratio (LR) and inference based on asymptotic normality of the MLE. These two forms of likelihood-based inference are asymptotically equivalent (Section 12.5) in the sense that they lead to the same conclusions for sufficiently large sample sizes. However, in real situations there can be a non-negligible difference between these two approaches (Section 4.3).

Using the likelihood ratio approach in the context of Example 1.1, an interval of plausible values of the unknown parameter p_0 is obtained as all values p for which the log-likelihood is above a certain threshold. In Section 3.4 it is shown that the threshold can be chosen so that the resulting interval has desirable frequentist properties. In the continuation of Example 1.1 below, the threshold is chosen so that the resulting interval is a (approximate) 95 % confidence interval for parameter p.

The curvature of the log-likelihood is of fundamental importance in both the theory and practice of likelihood inference. The curvature is quantified by the second derivative, that is, the change in slope. When evaluated at the MLE, the second derivative is negative (because the slope changes from being positive for $p < \widehat{p}$ to negative for $p > \widehat{p}$) and the larger its absolute value the more sharply curved the log-likelihood is at its maximum. Intuitively, a sharply curved log-likelihood is desirable because this narrows the range over which the log-likelihood is close to its maximum value, that is, it narrows the range of plausible parameter values. In Section 3.2 it is seen that the variance of the MLE can be estimated by the inverse of the negative of the second derivative of the log-likelihood. This is particularly convenient in practice because some optimization algorithms evaluate the second derivative of the objective function as part of the algorithmic calculations (see Section 5.2). In the maximum likelihood context, the objective function is the log-likelihood, and the estimated variance of the MLE is an easily-calculated byproduct from such optimizers. The approximate normality of MLEs enables confidence intervals and hypothesis tests to be performed using well-established techniques.

The likelihood ratio and curvature-based methods of likelihood inference are demonstrated in the following continuation of Example 1.1.

Example 1.1 continued. The log-likelihood function for $p,\ 0 < p < 1$, is

$$\begin{aligned} l(p) &= \log L(p) \\ &= \log\left(\frac{100!}{90!\,10!}\right) + 10\log p + 90\log(1-p) \\ &= 30.48232 + 10\log p + 90\log(1-p)\,, \end{aligned} \tag{1.2}$$

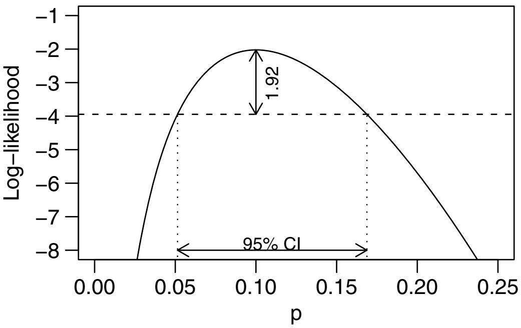


Figure 1.2 Binomial log-likelihood for 10 successes from 100 trials, and 95 % likelihood ratio confidence interval.

and the maximized value of this log-likelihood is $l(\widehat{p}) = l(0.1) \approx -2.03$.

In Section 3.4 it is seen that an approximate 95 % likelihood ratio confidence interval for parameter p is given by all values p_0 for which $l(p_0)$ is within about 1.92 of the maximized value of the log-likelihood. (The value 1.92 arises as one half of the 0.95 quantile of a chi-square distribution with one degree of freedom.) So, in this case, the interval is given by all values of p_0 for which $l(p_0)$ is -3.95 or higher. The confidence interval can be read from Figure 1.2, or obtained numerically for greater accuracy. This interval is $(0.051, 0.169)$ to the accuracy of three decimal places. From the equivalence between confidence intervals and hypothesis tests (Section 13.2) it can be concluded that the null hypothesis $H_0 : p = p_0$ will be rejected at the 5 % level for any value of p_0 outside of the interval $(0.051, 0.169)$.

To perform inference based on the curvature of the log-likelihood, the second derivative of the log-likelihood is required. This second derivative is given in Equation (11.15), and for $n = 100$ trials and $y = 10$ successes it is

$$l''(p) = \frac{\partial^2 l(p)}{\partial p^2} = -\frac{10}{p^2} - \frac{90}{(1-p)^2} \, . \tag{1.3}$$

Evaluating this second derivative at the MLE $\widehat{p} = 0.1$ gives

$$l''(\widehat{p}) = -\frac{10}{0.01} - \frac{90}{0.81} \approx -1111.111 \, .$$

The inverse of the negative of $l''(0.1)$ is exactly 0.0009, and according to likelihood theory (Sections 3.2 and 12.2), this is the approximate variance of $\widehat{p}$. The approximate standard error is therefore $\sqrt{0.0009} = 0.03$.

Recall that for a binomial experiment, the true variance of $\widehat{p}$ is $p_0(1 - p_0)/n$, which is estimated by $\widehat{p}(1 - \widehat{p})/n$. This estimate of variance is also 0.0009, the same as that obtained from using $-1/l''(0.1)$. (In fact, for the binomial the two variance estimates are always the same, for any values of n and y.)

For sufficiently large n, the distribution of $\widehat{p}$ can be approximated by a normal distribution, thereby permitting approximate tests and confidence intervals for p to be performed using familiar techniques. These are often called Wald tests or intervals, due to the influential work of Abraham Wald in establishing the large-sample approximate normality of MLEs (e.g. Wald 1943). The $(1-\alpha)100\,\%$ Wald confidence interval for p can be obtained using the familiar formula that calculates the upper (or lower) bounds as the point estimate plus (or minus) $z_{1-\alpha/2}$ times the estimated standard error ($\widehat{\text{se}}$), where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution. Thus, the approximate 95 % Wald confidence interval is

$$\widehat{p} \; \pm z_{0.975} \; \widehat{\text{se}}(\widehat{p}), \tag{1.4}$$

where $z_{0.975} \approx 1.96$ and $\widehat{\text{se}}(\widehat{p}) = 0.03$. This interval is (0.041, 0.159). Equivalently, this interval is the collection of the values of p_0 such the null hypothesis $H_0 : p = p_0$ is not rejected at the 5 % level by the Z-statistic. This is the values of p_0 that satisfy the inequality

$$|Z| = \left| \frac{\widehat{p} - p_0}{\widehat{\text{se}}(\widehat{p})} \right| < z_{0.975} \; . \tag{1.5}$$

□

> Although the Wald CI and test statistic in (1.4) and (1.5) may be the most commonly taught and used methods of such inference for the binomial model, it is hoped that this text will convince the reader to avoid Wald (i.e. approximate normality) methodology whenever it is practicably feasible. See the next section for more on this.

Box 1.2

1.2.2 Approximate normality versus likelihood ratio

The Wald form of confidence interval used in (1.4) is based on the approximate normal distribution of $\widehat{p}$. This is the most commonly used method for constructing approximate confidence intervals because of its intuitive appeal and computational ease. It was shown earlier that the likelihood ratio can be used as an alternative method for constructing confidence intervals – which should be used?

From a pragmatic point of view, there is considerable intuitive appeal in the Wald construction of a 95 % (say) confidence interval, with bounds given by 1.96 standard errors each side of the point estimate. This form of CI will be the most familiar to anyone with a basic grounding in frequentist statistics. However, when the LR and Wald intervals differ substantially, it is generally the case that the LR approach is superior, in the sense that the CIs obtained using likelihood ratio will have actual coverage probability closer to the a priori chosen value of (1-α) (see

Section 4.3.1). In fact, the results of Brown *et al.* (2001) question the popular usage of the Wald CI for binomial inference because of its woeful performance, even for some values of n and p for which the normal approximation to the binomial distribution is generally considered reasonable (typically, $\min(n\widehat{p}, n(1-\widehat{p})) \geq 5$). Unfortunately, the LR confidence interval is not as widely used because it requires (a little) knowledge of likelihood theory, but more importantly because it can not generally be calculated explicitly.

Application of Wald tests and construction of CIs extends to multi-parameter inference, but becomes more cumbersome and unfamiliar when simultaneous inference about two or more parameters is required. It is then that LR-based inference tends to be more commonly used. In particular, multi-parameter inference is typical of model selection problems, and in this area LR-based inference dominates. Also, it should be noted that model selection criterion such as Akaike's Information Criterion (AIC) (Section 4.4.1) make direct use of the likelihood.

In addition to the Wald and LR intervals, there are several other competing methods for constructing approximate confidence intervals for the probability parameter p in a binomial experiment. These include the Wilson score (see Box 3.1, Example 12.10, and Exercise 12.7), Agresti-Coull, and the misnamed 'exact' CIs. The comparisons performed by Agresti and Coull (1998) and Brown *et al.* (2002) suggest that the LR and Wilson score CIs are to be preferred.

Box 1.3

Summary

To conclude, Example 1.1 demonstrates likelihood inference in a nutshell. Much of the rest of this book is devoted to providing pragmatic guidance on the use (and potential abuse) of inferential methods based on likelihood ratios and approximate normality of MLEs, and their application to more complex and realistic models. These concepts extend naturally to models with two or more parameters, although the implementation can become challenging. For example, in a model where the number of parameters is $s > 2$, the second derivative of the log-likelihood is an s-dimensional square matrix (the Hessian) and the negative of its inverse provides an approximate variance matrix for the MLEs.

1.3 Using SAS, R and ADMB

This book is not just about understanding maximum likelihood inference, it is also very much about *doing* it with real data. Examples in SAS and R (Ihaka and Gentleman 1996, R Development Core Team 2010) are provided throughout Part II, along with a smattering of examples demonstrating Automatic Differentiation Model Builder (ADMB, ADMB-project (2008a, or any later version)).

Unlike the SAS and R environments, ADMB is a tool specifically designed for complex optimization problems. Due to the learning curve required to use ADMB, its use is difficult to justify if existing functionality within SAS or R can be used instead. Other than the quick demonstration of ADMB later in this chapter, it is used sparingly until Chapter 10 where it becomes the best choice for the general-purpose fitting of latent variable models. Some of its additional capabilities are noted in Sections 4.2.3 and 5.4.2.

The SAS examples presented in this text were implemented using SAS for Windows version 9.2. The SAS procedures used throughout are found in the statistics module SAS/STAT (SAS Institute 2008), with the exception that occasional use was made of the nonlinear optimizer `PROC NLP` which is in the operations research module SAS/OR. Some users of SAS/STAT may find that their licence does not extend to SAS/OR and hence will not be able to use `PROC NLP`. For this reason, `PROC NLP` is used sparingly and alternative SAS code is given where possible.

SAS procedures typically produce a lot of output by default. The output often includes a lot of superfluous information such as details about the contents of the data-set being used, computational information, and unwanted summary statistics. Throughout, the Output Delivery System (ODS) in the SAS software has been used to select only the required parts of the output produced by the SAS procedure.

Delwiche and Slaughter (2003, or any later edition) provides an excellent introduction to SAS. For ease of readability, the SAS code presented herein follows their typographical convention. This convention is to write SAS keywords in uppercase, and to use lowercase for variable names, data-set names, comments, etc. Note that SAS code is not case sensitive.

The R examples were run using R for Windows version 2.12.0. R is freely available under the terms of the Free Software Foundation's GNU General Public License (see `http://www.R-project.org`. Most of the R functions used herein are incorporated in the default installation of R. Others are available within specified R library packages, and can be easily loaded from within the R session.

ADMB is freely available via the ADMB project (`http://www.admb-project.org`), where full instructions for using ADMB can also be found. A short description of automatic differentiation is given in Section 15.6. In brief, ADMB is implemented by programming the objective function within an ADMB template file. The objective function is just the (negative) log-likelihood (and in latent variable models the density function of the latent variables also needs to be specified). An executable file is then created from the template file. Fortunately, much of the detail in creating the executable can be hidden behind convenient user interfaces. The ADMB examples in this book were run from within R using the interface provided by the `PBSadmb` package.

In many applications of ML inference it will be possible to make use of existing SAS procedures and R functions that are appropriate to the type of data being modelled, notwithstanding that this convenience often comes at the loss of flexibility. Rather than using existing functionality that is specific to the binomial model, the implementations of Example 1.1 presented below demonstrate a selection of the general-purpose tools available in SAS and R, and the use of ADMB. In particular, calculation of likelihood ratio confidence intervals is an application of profile likelihood (Section 3.6), and the examples below make use of general-purpose code for this purpose.

1.3.1 Software resources

Several small pieces of code have been written to facilitate techniques described in this text. These are listed in Section 15.5, along with a brief description of their functionality. These software resources are freely available for download from `http://www.stat.auckland.ac.nz/~millar`. This web resource also contains the complete code, and the data, for all examples used in this text.

1.4 Implementation of the motivating example

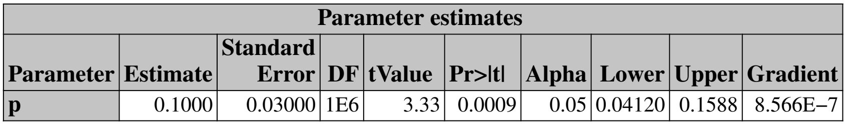
The code used below demonstrates how an explicit log-likelihood function is maximized within each of SAS, R and ADMB, and the calculation of the Wald and likelihood-ratio confidence intervals. Some efficiencies could have been gained by taking advantage of built-in functionality within the software. For example, in the SAS example, the binomial model could have been expressed using the statement `MODEL y ~ BINOMIAL(n,p)`, but the general-purpose likelihood specification has been used here for illustration. In R, various functionality (e.g. the `mle` function in package `stat4`, or `maxLik` function in the package of the same name) could have been used to shortcut some of the required code. However, the savings are minimal, and it is instructive to see the individual programming steps.

The first term of the binomial log-likelihood given in Equation (1.2) is a constant, and hence is irrelevant to maximization of the log-likelihood. However, it is good practice always to include the constant terms because it removes possible sources of confusion when fits of different model types are being compared (e.g. using Akaike's information criterion), or when verifying the fit of a model by using an alternative choice of software. Inclusion of the constant terms in the log-likelihood is becoming standard in most software applications of ML, but do not take this for granted.

The description of the code that is presented below is relatively complete, but this level of explanation is too unwieldy to be used throughout the remainder of this text. For more explanation on programming details and syntax, the reader should refer to the abundant online resources and documentation for each of these software.

1.4.1 Binomial example in SAS

The SAS code below uses `PROC NLMIXED` to implement Example 1.1, and produces the output shown in Figure 1.3.



Parameter estimates									
Parameter	Estimate	Standard Error	DF	tValue	Pr>|t|	Alpha	Lower	Upper	Gradient
p	0.1000	0.03000	1E6	3.33	0.0009	0.05	0.04120	0.1588	8.566E−7

Figure 1.3 The parameter estimates table from `PROC NLMIXED`*, including the 95 % Wald confidence interval (0.0412,0.1588).*

```
DATA binomial;
  y=10; n=100;
RUN;

*Select only the parameter estimates table;
ODS SELECT ParameterEstimates;

PROC NLMIXED DF=1E6 DATA=binomial;
  PARMS p=0.5;
  BOUNDS 0<p<1;
  loglhood=LOG(COMB(n,y))+y*log(p)+(n-y)*log(1-p);
  MODEL y~GENERAL(loglhood);
RUN;
```

Some features of the above code are:

- The default output includes several tables, including tables of log-likelihood values and fit statistics. The Output Delivery System statement `ODS SELECT ParameterEstimates;` is used to select only the required table.
- By default, `NLMIXED` calculates Wald intervals using a t-distribution with degrees of freedom equal to the number of observations (rows in the dataset). To get the normal-based Wald interval in (1.4), the value for the degrees of freedom needs to be set to a large number. In this case, it was set to one million using the procedure option `DF=1E6`.
- The `PARMS` statement is an optional statement used to explicitly list the parameters and their initial values.
- The `BOUNDS` statement is an optional statement used to specify the range of the parameter values (i.e. the parameter space).
- The model is specified using the `MODEL` statement. Here, the model is given as `GENERAL(loglhood)` to specify that `PROC NLMIXED` should maximize the value of the log-likelihood, `loglhood`, as specified by the preceding programming statement.
- In the SAS output in Figure 1.3, `Gradient` gives the slope of the log-likelihood upon termination of the optimization. It should be near zero. If not, then convergence of the optimizer to a maximum of the log-likelihood may not have been achieved.

- The `t-Value` and `Pr>|t|` columns in Figure 1.3 should be ignored. They are the Wald test statistic and p-value for the null hypothesis $H_0 : p = 0$. This is not a relevant hypothesis here.

One current limitation (in SAS 9.2) is that `PROC NLMIXED` does not produce likelihood ratio confidence intervals. A general-purpose macro called `Plkhci` has been written for this purpose.

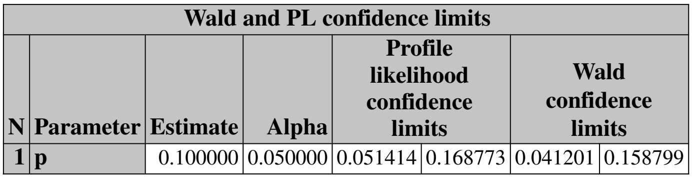
```
%INCLUDE "PlkhciMacro.sas";

%MACRO BinomialProfile(p);
  PROC NLMIXED DF=1E6 DATA=Binomial; TECH=NONE;
    loglhood=LOG(COMB(n,y))+y*log(&p)+(n-y)*log(1-&p);
    MODEL y~GENERAL(loglhood);
  RUN;
%MEND;
%Plkhci(BinomialProfile,0.0,0.1,-2.0259739,side="L");
%Plkhci(BinomialProfile,0.1,1.0,-2.0259739,side="R");
```

- The user-defined macro `BinomialProfile` contains a modified version of the `NLMIXED` code that was used to produce the output in Figure 1.3, and this is passed as an argument to the profile likelihood macro `Plkhci`. More description of these macros is found in Sections 3.4.1 and 15.5.3. Note that macro commands are specified using the `%` symbol.
- The `Plkhci` macro finds the likelihood ratio confidence bounds. It writes the following lines to the log window of the SAS session:

```
Left-sided 95% LR CI bound is 0.051413
Right-sided 95% LR CI bound is 0.168779
```

For SAS installations that include the operations research OR module, `PROC NLP` provides an easier option for obtaining the likelihood ratio confidence interval, via its `PROFILE` statement. Figure 1.4 shows the table that is produced from running the following code.

```
*Select only the desired table;
ODS SELECT WaldPLLimits;
PROC NLP COV=2 VARDEF=N;
  MAX loglhood;
  PROFILE p / alpha=0.05;
  PARMS p=0.5;
  BOUNDS 0<p<1;
  n=100; y=10;
  loglhood=LOG(COMB(n,y))+y*LOG(p)+(n-y)*LOG(1-p);
RUN;
```



Wald and PL confidence limits

N	Parameter	Estimate	Alpha	Profile likelihood confidence limits		Wald confidence limits	
1	p	0.100000	0.050000	0.051414	0.168773	0.041201	0.158799

Figure 1.4 Likelihood ratio and Wald confidence limits from `PROC NLP`.

- `PROC NLP` provides a choice of several different estimates of variance and the option `COV=2` specifies use of the curvature-based estimate employed in the motivating example. Also, by default, `PROC NLP` makes a degrees-of-freedom adjustment to the estimate of variance. This adjustment is not appropriate in the maximum likelihood context, and the procedure option `VARDEF=N` prevents this.

- The `MAX loglike` statement specifies that the value of `loglike` is to be maximized.

- The `PROFILE` statement requests calculation of a likelihood ratio confidence interval for parameter p, with confidence level $(1 - \alpha)100\,\%$.

1.4.2 Binomial example in R

The R code presented below uses the general-purpose minimizer `optim`, and hence the objective function to be minimized is the negative of the log-likelihood. This is explicitly defined as function `nloglhood`, with argument p. The likelihood ratio confidence interval is obtained using the `plkhci` function (from the `Bhat` package) for profile likelihood confidence intervals.

```
> #Define the negative log-likelihood function
> nloglhood=function(p)
+   return( -(log(choose(100,10))+10*log(p)+90*log(1-p)) )
> #Minimize the negative log-likelihood
> binom.fit=optim(0.5,nloglhood,lower=0.0001,upper=0.9999,
+                 hessian=T)
> phat=binom.fit$par #The MLE
> phat.var=1/binom.fit$hessian #Variance is inverse hessian
> #Calculate approximate 95% Wald CI
> phat+c(-1,1)*qnorm(0.975)*sqrt(phat.var)
[1] 0.04120779 0.15879813

> library(Bhat) #Loading package Bhat
> #Set up list for input into plkchi function
> control.list=list(label="p",est=phat,low=0,upp=1)
> #Calculate approximate 95% likelihood ratio CI
> plkhci(control.list,nloglhood,"p")
[1] 0.05141279 0.16877909
```

- In the call of `optim`, the first argument specifies that the initial parameter value to be used by the optimizer is 0.5. The `lower` and `upper` arguments specify the parameter space – in this case they were set to 0.0001 and 0.9999 because computational error occurred if bounds of 0 and 1 were used due to `nloglhood` being undefined at these values. The `hessian=T` argument requests that the value of the second derivative of the negative log-likelihood (calculated at the MLE) be included in `binom.fit`.
- The list object `binom.fit` has several components, including the value of the calculated MLE, `binom.fit$par`, and the second derivative of the negative log-likelihood, `binom.fit$hessian`.
- The first argument to the profile likelihood function `plkhci` is a list with elements giving the parameters of `nloglhood`, the MLE, and lower and upper bounds of the parameter space.

1.4.3 Binomial example in ADMB

The following ADMB template file, `BinomialMLE.tpl`, is used to find the MLE and its approximate standard error.

```
DATA_SECTION
  init_number y
  init_number n

PARAMETER_SECTION
  init_bounded_number p(0,1)
  objective_function_value nloglhood

PROCEDURE_SECTION
  nloglhood=-(lgamma(n+1)-lgamma(y+1)-lgamma(n-y+1)); // Constant term
  nloglhood=nloglhood-(y*log(p)+(n-y)*log(1-p));
```

- ADMB requires a data section, and the data are contained in a file with name `BinomialMLE.dat`. This text file contains a single row, the contents of which is `10 100`.
- The parameter section specifies the parameter and that it is bounded between 0 and 1. It also specifies the variable that will contain the value of the negative log-likelihood, and this is calculated in the procedure section.
- The `lgamma` function is the log-gamma function, and for non-negative integers x, `lgamma`$(x + 1) = \log(x!)$.

An executable program is generated from the template file, and executed, using the following R code. This uses functions from the `PBSadmb` package.

```
library(PBSadmb)
readADopts()
makeAD("BinomialMLE")
runAD("BinomialMLE")
```

- readADopts reads a text file containing information about the ADMB installation.
- makeAD creates a C++ file from the template file, compiles it, and links it to produce an executable with name BinomialMLE.exe.
- runAD runs the executable. A number of files are produced, including text file BinomialMLE.std containing the MLE and its standard error, and text file BinomialMLE.par containing the value of the negative log-likelihood at the MLE.

A few additional lines of code are required to obtain the likelihood ratio confidence interval. The parameter section requires addition of a likeprof_number specification to name the quantity of interest. In this case it is *p*, but this name is already in use, so the variable pcopy is used to copy the value of *p*. The preliminary calculations section is used to set options for the grid of pcopy values over which the objective function is evaluated.

```
DATA_SECTION
  init_number y
  init_number n

PARAMETER_SECTION
  init_bounded_number p(0,1)
  objective_function_value nloglhood
  likeprof_number pcopy

PRELIMINARY_CALCS_SECTION
  pcopy.set_stepnumber(500);
  pcopy.set_stepsize(0.01);

PROCEDURE_SECTION
  nloglhood=-(lgamma(n+1)-lgamma(y+1)-lgamma(n-y+1)); // Constant term
  nloglhood=nloglhood-(y*log(p)+(n-y)*log(1-p));
  pcopy=p;
```

Within R, the executable is created as before. The runAD function now requires the optional lprof argument to pass to the executable, to force calculation of the likelihood ratio confidence interval. If the template file is named BinomialLRCI.tpl, then the runAD call looks like

```
runAD("BinomialLRCI",argvec="-lprof > RunWindow.txt")
```

which also redirects a copious log file into the text file `RunWindow.txt`. This produces a text file `pcopy.plt`, in which the bounds of the 95 % confidence interval, calculated to be `0.0514` and `0.1687` can be found.

1.5 Exercises

1.1 The Poisson distribution is the default distribution for the modelling of count data. If the value $y = 3$ is observed from a Poisson distribution with unknown parameter λ then the log-likelihood is $l(\lambda) = -\lambda + 3\log\lambda - \log(6)$. This log-likelihood is maximized by $\widehat{\lambda} = 3$.

1. Plot $l(\lambda)$ for values of λ from 0.1 to 15.
2. By suitable modification to the program code in Section 1.3, use R, SAS or ADMB to verify that $\widehat{\lambda} = 3$, and to calculate the 95 % Wald and likelihood ratio confidence intervals for λ.