Proceedings of the 2009 Winter Simulation Conference
M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, eds.

RESAMPLING METHODS OF ANALYSIS IN SIMULATION STUDIES

Russell Cheng
Christine Currie

School of Mathematics
University of Southampton
Southampton, SO17 1BJ, UK

ABSTRACT

This is an introductory tutorial on the statistical analysis of simulation output, but focusing on the (elementary) use of resampling, and related computer intensive techniques. The aspects covered are (i) input modeling (ii) output analysis (iii) model validation and (iv) model building and selection. The presentation will be very practically oriented including a fair number of real-time spreadsheet demonstrations. The demonstration worksheets will be made freely available online, and participants are actively encouraged to download them to try out the methods in their own simulations.

1 INTRODUCTION

This tutorial aims to bring together modeling and statistical methodology in the way that it ought to be used in practice. The tutorial provides a viewpoint that we have found most useful in tackling a modeling problem - the sort of things that in retrospect we wished had been pointed out to us when we first encountered problems of this sort.

Most simulationists will already have encountered most of the methods to be discussed. However these may have been encountered in a piecemeal, ad hoc way. Thus the way that these methods invariably come together when studying a modeling problem may not have been made completely apparent, so that their power may not be fully appreciated.

We therefore revisit statistical and modeling methodology in a way that emphasizes how they ought to be used in practice, discussing what should be going through the mind of the investigator at each stage. We show that an overall problem can be broken down into a standard set of subproblems all of which will invariably occur in a particular order. The subproblems will be reviewed and discussed, in a unified way in the tutorial.

A very good book that has a similar philosophy to this tutorial is An Introduction to Statistical Modelling by W.J. Krzanowski, (1998) Arnold, London. However this reference has a stronger statistical emphasis than we adopt and gives rather less attention to the resampling methods that we shall be using in the analysis.

Resampling is quite well covered in the book Computer Intensive Statistical Methods by J.S.U. Hjorth (1994) Chapman & Hall, London. One problem with this reference is the order in which material is presented. The initial chapters deal with arguably somewhat advanced topics. A good starting point for the book is Chapter 5.

The tutorial includes web links to a number of Excel workbook examples containing VBA macros for carrying out the statistical calculations discussed in the text. For reasons of space discussion of these workbooks has had to be kept to a minimum in this article. However the workbooks themselves contain extra clarifying guidance on how they can be used. Moreover the macros themselves are annotated to assist anyone who wishes to modify them to extend their range of application. The authors welcome comments or suggestions concerning the usefulness or otherwise of these workbooks.

2 STATISTICAL METAMODELS

We emphasize the importance of statistical *metamodels* for analysing data. We need therefore to be clear what is meant by a metamodel and this is discussed first.

Figure 1 illustrates the situation where we have data, **Y** (here and throughout this text, a quantity is written in bold to indicate that it is a vector quantity), available concerning the behavior of a system under study. The system itself, represented

by the box in the middle, might be simple but it will typically be complicated or even unknown. We call **Y** the *output* and this is what we wish to analyze, to learn about the behavior of the system.

We also have *input* quantities, whose values are expected to influence the output. The inputs are divided into two types. The input $\mathbf{X} = (X_1, X_2, \ldots, X_M)$ is a vector of M explanatory variables. These are known quantities and indeed may possibly be under the control of the investigator. The input $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_p)$ is a vector of parameters which influence the output but whose values are not controllable. Often they will be unknown. Their values would therefore have to be *estimated* using data or past information, **w**, containing information about $\boldsymbol{\theta}$. We write these estimates as $\hat{\boldsymbol{\theta}}(\mathbf{w})$, or simply as $\hat{\boldsymbol{\theta}}$. Figure 1 includes this possibility.

In addition the output **Y** may contain a random component, typically referred to as 'noise' or 'error'. This is denoted by $\boldsymbol{\varepsilon}$.

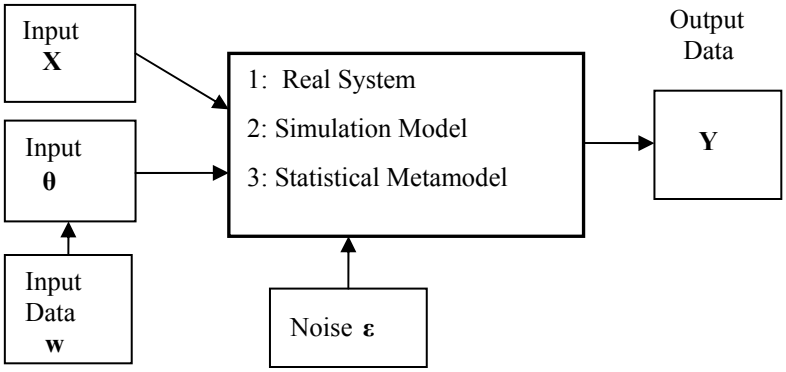


Figure 1: Schematic showing the similar structure of a real system, a simulation model and a statistical metamodel.

As well as depicting the situation where the output data has been obtained from a real system Figure 1 also illustrates the situation where we have constructed a simulation model and have made simulation runs with it to obtain simulated output data. This is indicated in Figure 1 by replacing the real system in the central block by a simulation model. All other blocks remain the same.

In this tutorial the focus is on how to analyze **Y** and in particular to identify how the inputs **X** and $\boldsymbol{\theta}$ influence **Y** in the presence of the random effects $\boldsymbol{\varepsilon}$. We use a *statistical model* for doing this. We shall make precise later what is meant by a statistical model. However we observe here that the structure of the process is unchanged, and this is emphasized by using Figure 1 yet again, only with the central block now representing the statistical model.

The term statistical model is conventionally used when we are analyzing data obtained from a real system. In the case of data obtained from a simulation model, then the statistical model is a model of a model, so to speak – and this is when the term *metamodel* is used. It will be clear that whatever statistical model is deemed appropriate in a given situation is determined purely by the structure of the data and not by its origin. Thus the model would apply whether the output came from a real system or a simulation model.

The following is a typical example of a statistical model. Here, and in the rest of the paper, we use the symbol '□' to mark the end of an example.

Example 1: Consider the operation of a single server queue where we are interested in estimating the long term average queue length. Here Y might be the sampled mean queue length over a period of length T. Input quantities are λ the arrival rate, and μ, the service rate. Typically λ might be treated as an explanatory variable, possibly not under our control, whilst μ might be a known input. This is a situation that has been well analyzed theoretically and the long term average queue length, is known to depend only on the so-called *traffic intensity* $\rho = \lambda / \mu$. In the spreadsheet example, `<Traffic Queue Length EG>`, n independent observations have been obtained of Y and ρ, each over a time period of length T. If T is sufficiently large we might therefore assume that the data take the form

$$y_i = \eta(\rho_i) + \varepsilon_i, \; i = 1,2,\ldots,n, \tag{1}$$

where $\eta(\rho)$ is some suitably selected function characterizing the likely behavior of Y. The quantity ε is a random variable. A common assumption is that the errors have a normal distribution:

$$\varepsilon \sim N(0,\sigma^2)\,. \tag{2}$$

This assumption of constant variance is very dubious in the present context as the variability of Y is known to depend heavily on ρ. It would therefore be better to allow the variance to depend on ρ □

The form (1) can be regarded as a particular case of a *regression* where

$$Y = \eta(\mathbf{X},\boldsymbol{\theta}) + \varepsilon$$

with

$$E(Y) = \eta(\mathbf{X},\boldsymbol{\theta})$$

called the *regression function*. Here the inputs $\mathbf{X}$ are treated as *explanatory variables* on which the regression function depends, whilst $\boldsymbol{\theta}$ are treated as parameters on which the function also depends. In dynamic problems one of the explanatory variables is usually *time*.

The regression function $\eta(\mathbf{X}, \boldsymbol{\theta})$ may have to be selected so that its behavior resembles the output of the system or mathematical/simulation model that it represents. In some situations the physical process of the actual system may be sufficiently known to suggest a natural form for $\eta(\mathbf{X}, \boldsymbol{\theta})$.

If little is known about the real system, the form assumed for $\eta(\mathbf{X}, \boldsymbol{\theta})$ does not have to be complicated. When there is a single explanatory variable X then a low polynomial function of X is a typically used model:

$$\eta(X,\boldsymbol{\theta}) = \beta_0 + \beta_1 X + \ldots + \beta_M X^M\,.$$

When there are a large number of factors, and especially when the errors $\boldsymbol{\varepsilon}$ are not small then a multivariate linear form is often used:

$$\eta(X,\boldsymbol{\theta}) = \beta_0 + \beta_1 X_1 + \ldots + \beta_M X_M\,.$$

Here the X_i are the values of the different factors, and the model only considers the inclusion of a linear term for each factor.Example 2 is an illustration of a situation where this multivariate form is appropriate.

Example 2: In the study of a supply chain, Kleijnen, Bettonvil, and Persson (2006) discuss screening techniques in a simulation study involving 92 factors. Here we consider data collected from a set of 128 simulation runs (using a Plackett-Burman design) given in the following link: <Ericsson EG>. The outputs Y are steady-state mean costs and we use the well known *multiple linear regression* model

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_k X_{iM} + \varepsilon_i,\ i = 1,2,\ldots,n \tag{3}$$

as a simple exploratory model to identify which factors are important in determining the cost. (For simplicity the example is set up initially for just 15 factors, but the data for all 92 factors is provided in the workbook.) Here Y_i is the observed cost in the ith simulation run, and X_{ij} is the observed value of the jth factor in the ith run; and we have n runs. Again we might assume normal errors

$$\varepsilon \sim N(0,\sigma^2)\,.$$ □

Example 3: A simulation queueing model was used in a study of the Severn River Toll Bridge in the UK. Toll service times were modeled using the two parameter gamma distribution $G(\alpha,\beta)$. The parameters were estimated from observed toll service times. The spreadsheet <Gamma MLE> contains a typical sample of 47 observed service times (in seconds). □

We will be focusing on the third representation of Figure 1 where we use a *statistical model* to describe the output. The first step in model formulation is therefore to write down the distributional form of the output and in particular to make ex-

plicit how the distribution is expected to depend on the input quantities. Often the regression format is a convenient one to use.

The procedure of treating Y as a random variable and writing down its distribution by *name*, is a very good one to follow. The distribution will usually depend on parameters. It is also necessary therefore to write down how these parameters of the distribution depend on the input variables and on the input parameters of the process model. *This first step of treating the output Y as a random variable and of identifying its distribution is essential in determining the most appropriate subsequent analysis.*

We discuss the main characteristics of random variables in the next Section.

3 RANDOM VARIABLES

The key concept of all statistics is the *random variable*. We avoid a formal definition but treat a random variable simply as a quantity that one can observe many times but that takes different values each time it is observed in an unpredictable, random way. These values however will follow a *probability distribution*. The probability distribution is therefore the defining property of a random variable. Thus, given a random variable, the immediate and only question one can, and should *always ask* is: *What is its distribution*?

We denote a random variable by an upper case letter X (Y, Z etc). An *observed value* of such a random variable will be denoted by a lower case letter x (y, z etc). The definition of most statistical probability distributions involves *parameters*. Well known parametric probability distributions are the normal, exponential, gamma, binomial and Poisson.

We focus on continuous distributions which are defined by their *probability density functions* (PDF), typically written as $f(y)$. The PDF is *not* a probability, however it can be used to form a *probability increment*. $\Pr\{y \le Y \le y + \delta y\} = f(y)\delta y$. This is a good way to view the PDF.

Example 4a: Write down the PDF of

(i) the normal distribution, $N(\mu, \sigma^2)$. <Normal Distribution>

(ii) the gamma distribution, $G(\alpha, \beta)$. <Gamma Distribution> □

An alternative way of defining a probability distribution is to give its *cumulative distribution function* (CDF), where the CDF, $F(x)$ gives the probability of the random variable X taking on a value of less than or equal to x.

Example 4b: Plot the CDF's of each of the random variables in the previous example. Hint: Use the Worksheet Functions NormDist and GammaDist. (These worksheet functions are used in later examples.) □

Example 4c: Generate samples of normal variates, and gamma variates using the *inverse transform method* (See Law and Kelton, 1991), and plot their frequency histograms. Here is a link to a workbook for doing this: <Normal/Gamma Variate generator> □

4 FITTING PARAMETRIC MODELS TO RANDOM SAMPLES: INPUT MODELLING

Random samples are the simplest data sets that are encountered. A random sample is just a set of *n independent and identically distributed* observations (of a random variable). We write it as $\mathbf{Y} = \{Y_1, Y_2, \ldots Y_n,\}$ where each Y_i represents one of the observations.

A basic problem is when we wish to fit a parametric distribution to a random sample. This problem is an elementary form of modeling called *input modeling*.

Example 3 (continued): This is a typical example of the input modeling problem. If we can estimate the parameters of the gamma distribution, we will have identified the toll booth service time distribution completely and can then use it to study the characteristics of the system, employing either queueing theory or simulation. □

To fit a distribution, a method of estimating the parameters is needed. The best method *by far* is the *method of maximum likelihood* (ML). The resulting estimates of parameters are called *maximum likelihood estimates* (MLEs). ML estimation is a completely general method that applies not only to input modeling problems but to all parametric estimation problems. We summarize the method next.

5 MAXIMUM LIKELIHOOD ESTIMATION

Suppose $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ is a set of observations where the ith observation, Y_i, is drawn from the continuous distribution with PDF $f_i(y, \boldsymbol{\theta})$ ($i = 1, 2, \ldots, n$). The subscript i indicates that the distributions of the Y_i can all be different. Then we write down the joint distribution of $\mathbf{Y}$ evaluated at the sampled value $\mathbf{y}$ as:

$$Lik(\boldsymbol{\theta}, \mathbf{y}) = f_1(y_1, \boldsymbol{\theta}) f_2(y_2, \boldsymbol{\theta}) \ldots f_n(y_n, \boldsymbol{\theta}) .$$

This expression, *treated as a function of* $\boldsymbol{\theta}$**,** is called the called the *likelihood* (of the sampled value $\mathbf{y}$)**.** The logarithm

$$L(\boldsymbol{\theta}, \mathbf{y}) = \log\{Lik(\boldsymbol{\theta}, \mathbf{y})\} = \log f_1(y_1, \boldsymbol{\theta}) + \log f_2(y_2, \boldsymbol{\theta}) + \ldots + \log f_n(y_n, \boldsymbol{\theta}) \tag{4}$$

is called the *loglikelihood*. *The ML estimate,* $\hat{\boldsymbol{\theta}}$*, is that value of* $\boldsymbol{\theta}$ *which maximizes the loglikelihood.* The MLE is illustrated in Figure 2 in the one parameter case.

In certain situations, and this includes some well known standard ones, the likelihood equations can be solved to give the ML estimators explicitly. This is preferable when it can be done. However in general the likelihood equations are not very tractable. Then a much more practical approach is to obtain the maximum using a *numerical search method*.

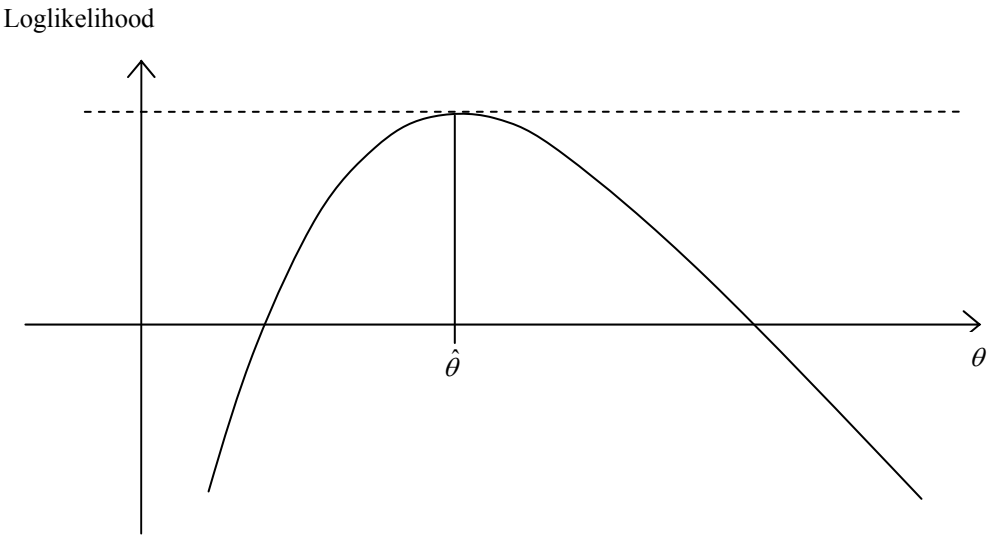


Figure 2: The maximum likelihood estimator $\hat{\theta}$

There exists a number of powerful numerical optimizing methods but these can be laborious to set up. An exception is the readily accessible numerical optimizer *Solver* which can be called from an Excel spreadsheet. This can handle problems that are not too large. A more flexible alternative is to use a direct search method like the *Nelder-Mead* method. A demonstration of this algorithm in action is provided here: `<Nelder-Mead>`.

Example 5 (Continuation of Example 3): The spreadsheet `<Gamma MLE>` contains a VBA subroutine using the Nelder-Mead method to fit the gamma distribution $G(\alpha, \beta)$ to data. Use it to fit this distribution to the toll booth data. □

6 ACCURACY OF MAXIMUM LIKELIHOOD ESTIMATORS

A natural question to ask of an MLE is: *How accurate is it?* Now an MLE, being just a function of the sample, is a statistic, and so is a random variable. Thus the question is answered once we know its distribution.

An important property of the MLE, $\hat{\boldsymbol{\theta}}$, is that its asymptotic probability distribution is known to be normal. The key working version of the result is that, for $n \to \infty$

$$\hat{\boldsymbol{\theta}} \sim N\{\boldsymbol{\theta}_0, \mathbf{V}(\hat{\boldsymbol{\theta}})\}, \tag{5}$$

where $\boldsymbol{\theta}_0$ is the unknown true parameter value and the variance-covariance matrix of the MLE

$$\mathbf{V}(\hat{\boldsymbol{\theta}}) \; '[-\partial^2 \mathbf{L}(\boldsymbol{\theta}, \mathbf{y}) / \partial \boldsymbol{\theta}^2 \big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}]^{-1}, \tag{6}$$

is the inverse of the negative of $\partial^2 \mathbf{L}(\boldsymbol{\theta}, \mathbf{y}) / \partial \boldsymbol{\theta}^2$, the second derivative of the loglikelihood. This latter is called the *Hessian* (of $\mathbf{L}$) and measures the *rate of change of the derivative* of the loglikelihood. This is essentially the *curvature* of the loglikelihood. Thus it will be seen that the variance is simply the inverse of the *magnitude* of this curvature at the stationary point.

Though easier to calculate than the *information matrix* $\mathbf{I}(\boldsymbol{\theta}) = E\left(-\partial^2 \mathbf{L} / \partial \boldsymbol{\theta}^2\right)$, which it replaces, the *observed information* $-\partial^2 \mathbf{L}(\boldsymbol{\theta}, \mathbf{y}) / \partial \boldsymbol{\theta}^2$ is often difficult to evaluate analytically. Again it is usually much easier to calculate it numerically using a finite-difference formula for the second derivatives. The expression is a matrix of course, and the variance-covariance matrix of the MLE is the negative of its *inverse*. A numerical procedure is needed for this inversion.

The way that (5) is typically used is to provide confidence intervals. For example a (1-α)100% confidence interval for the coefficient θ_1 is

$$\hat{\theta}_1 \pm z_{\alpha/2} \sqrt{V_{11}(\hat{\boldsymbol{\theta}})} \tag{7}$$

where $z_{\alpha/2}$ is the upper 100α/2 percentage point of the standard normal distribution.

Often we are interested not in $\boldsymbol{\theta}$ directly, but some arbitrary, but given function of $\boldsymbol{\theta}$, g($\boldsymbol{\theta}$) say. ML estimation has the attractive general *invariant* property that the MLE of g($\boldsymbol{\theta}$) is

$$\hat{g} = g(\hat{\boldsymbol{\theta}}). \tag{8}$$

An approximate (1-α)100% confidence interval for g($\boldsymbol{\theta}$) is then

$$g(\hat{\boldsymbol{\theta}}) \pm z_{\alpha/2} \sqrt{(\partial g / \partial \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}{}^T \mathbf{V}(\hat{\boldsymbol{\theta}}) (\partial g / \partial \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}}} \tag{9}$$

In this formula the first derivative of $g(\boldsymbol{\theta})$ is required. If this is not tractable to obtain analytically then, as with the evaluation of the information matrix, it should be obtained numerically using a finite-difference calculation.

Summarizing it will be seen that we need to

(i) Formulate a statistical model of the data to be examined. (The data may or may not have been already collected. The data might arise from observation of a real situation, but it might just as well have been obtained from a simulation.)

(ii) Write down an expression for the loglikelihood of the data, identifying the parameters to be estimated.

(iii) Use this in a numerical optimization of the loglikelihood, e.g. using Solver or Nelder Mead.

(iv) Use the optimal parameter values to obtain estimates for the quantities of interest.

(v) Calculate confidence intervals for these quantities.

Example 6: Suppose that the gamma distribution $G(\hat{\alpha}, \hat{\beta})$ fitted to the toll booth data of Example 5 is used as the service distribution in the design of an M/G/1 queue. Suppose the inter-arrival time distribution is known to be exponential with PDF

$$f(y) = \lambda e^{-\lambda y}, \quad y > 0.$$

but a range of possible values for the arrival rate, λ, needs to be considered. Under these assumptions the steady state mean waiting time in the queue is known to be

$$W(\lambda;\alpha,\beta) = \frac{\lambda(1+\alpha)\alpha\beta^2}{2(1-\alpha\beta\lambda)}. \tag{10}$$

Plot a graph of the mean waiting time $W(\lambda;\alpha,\beta)$ for the queue for $0 < \lambda < 0.1$ (per second), assuming that the service time distribution is gamma: $G(\hat{\alpha},\hat{\beta})$. Add 95% confidence intervals to this graph to take into account the uncertainty concerning α and β because estimated values $\hat{\alpha}$ and $\hat{\beta}$ have been used. Do this using the "PerformanceIndex" worksheet in <Gamma MLE>. □

The above formulas, based on the aymptotic theory of ML estimation, are useful but only become accurate with increasing sample size *n*. With existing computing power, *computer intensive methods* provide an excellent alternative. Experience (and theory) has shown these latter methods will often give better results in general for small sample sizes. Moreover this alternative approach is usually much easier to implement than the classical methodology. The method is called *bootstrap resampling*, or simply *resampling*.

Resampling hinges on the properties of the *empirical distribution function* (EDF) which we need to discuss first, and this is what we shall start with in the next section.

7 EMPIRICAL DISTRIBUTION FUNCTIONS

Consider first a single random sample of observations Y_i, for i = 1, 2, ..., n. The *empirical distribution function* (EDF) is defined as:

$$\tilde{F}(y) = \frac{\#\ of\ Y's \le y}{n}. \tag{11}$$

The EDF is illustrated in Figure 3. It is usually simplest to think of the observations as being *ordered*: $Y_{(1)} < Y_{(2)} < \ldots < Y_{(n)}$. These are what are depicted in Figure 3. Note that the subscripts are placed in brackets to indicate that this is an ordered sample.

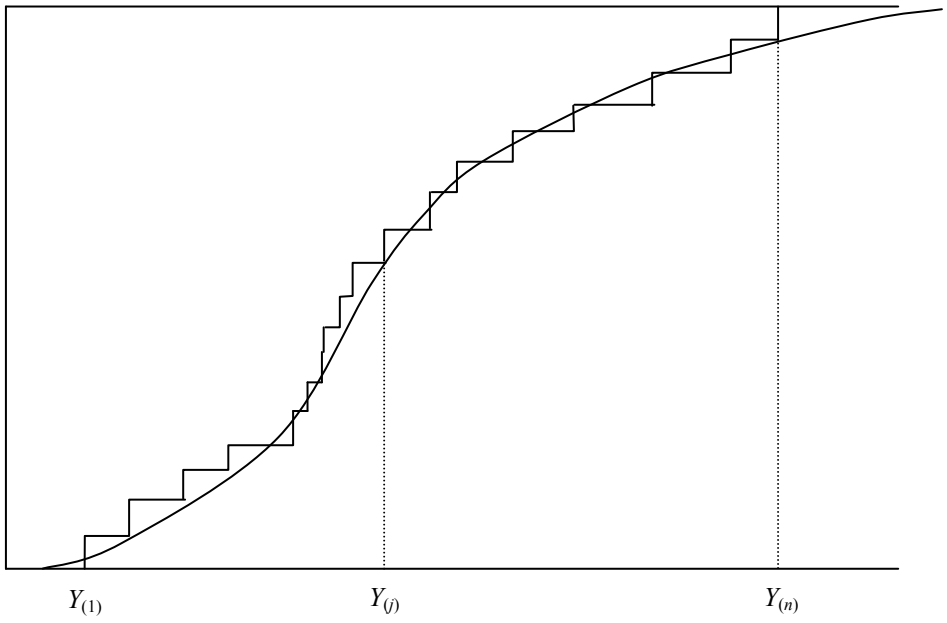


Figure 3: EDF of the Y_i, $\tilde{F}(y)$

The key point is that the EDF estimates the (unknown) *cumulative distribution function* (CDF) of *Y*. We shall make repeated use of the following:

Fundamental Theorem of Sampling: *As the size of a random sample tends to infinity then the EDF constructed from the sample will tend, with probability one, to the underlying cumulative distribution function (CDF) of the distribution from which the sample is drawn.*

(This result when stated in full mathematical rigor, is known as the Glivenko-Cantelli Lemma, and it underpins all of statistical methodology. It guarantees that study of increasingly large samples is ultimately equivalent to studying the underlying population.)

In the previous section we studied the CDF of the Y_i's of a random sample by fitting a parametric distribution and then studying the fitted parameters and the fitted parametric CDF. Using the EDF, we can do one of two things:

(i) We can study the properties of the Y_i directly using the EDF, without bothering to fit a parametric model at all.

(ii) We can use the EDF to study properties of the fitted parameters and fitted parametric distribution.

We shall discuss both approaches. We shall focus first on (ii) as we wish to utilize bootstrapping to give us an alternative way of studying the properties of MLE's to that provided by asymptotic normality theory, which was discussed in Section 6.

We postpone discussion of (i) until later. We simply note at this juncture that the attraction of using the EDF directly, rather than a fitted parametric CDF, is that we make no assumption about the underlying distributional properties of Y. Nor is it assumed to come from any particular family of distributions like the normal or Weibull. This flexibility is particularly important when studying or comparing the output from complex simulations where it is possible that the distribution of the output may be unusual. For example it may well be skew, or possibly even multimodal.

8 BASIC BOOTSTRAP METHOD

The basic process of constructing a given statistic of interest is illustrated in Figure 4. This depicts the steps of drawing a sample $\mathbf{Y} = (Y_1, Y_1, ..., Y_n)$ of size n from a distribution $F_0(y)$, and then the calculation of the statistic of interest T from $\mathbf{Y}$. The problem is then to find the distribution of T.

Bootstrapping is a very general method for numerically estimating the distribution of a statistic. *It is a resampling method that operates by sampling from the data used to calculate the original statistic.*

Bootstrapping is based on the following idea. Suppose we could repeat the basic process, as depicted in Figure 4, a large number of times, B say. This would give a large sample $\{T_1, T_2,..., T_B\}$ of test statistics, and, by the Fundamental Theorem of Section 3, the EDF of the T_i will tend to the CDF of T as B tends to infinity. Thus, not simply does the EDF estimate the CDF, it can be made *accurate to arbitrary accuracy* at least in principle, simply by making B sufficiently large.

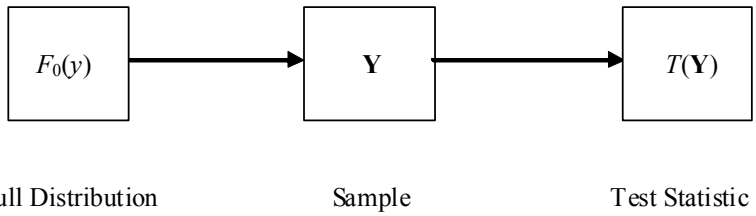


Figure 4: Basic sampling process

Unfortunately to apply this result requires repeating the basic process many times. In the present context this means having to repeat the simulation trials many times - something that is usually too expensive and impractical to do. The bootstrap method is based on the idea of replacing $F_0(y)$ by the best estimate we have for it. *The best available estimate is the EDF constructed from the sample* **Y.** Thus we mimic the basic process depicted in Figure 4 but instead of sampling from $F_0(y)$ we sample from the EDF of **Y**. This is exactly the same as sampling with replacement from **Y.** We carry out this process B times to get *B bootstrap samples* $\mathbf{Y}_1^*, \mathbf{Y}_2^*, ..., \mathbf{Y}_B^*$**.** (We have adopted the standard convention of adding an asterisk to indicate that a sample is a bootstrap sample.) From each of these bootstrap samples we calculate a corresponding *bootstrap statistic value* $T_i^* = T(\mathbf{Y}_i^*)$, $i = 1, 2, ..., B$. The process is depicted in Figure 5.

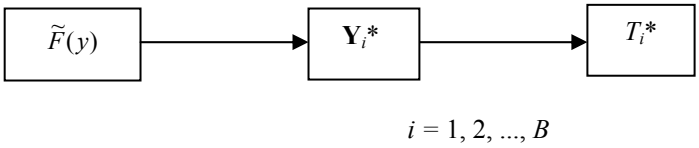


Figure 5: Bootstrap process

The EDF of the bootstrap sample, which without loss of generality we can assume to be reordered, so that $T_{(1)}^* < T_{(2)}^* < < T_{(B)}^*$, now estimates the distribution of T. This is depicted in Figure 6. Figure 6 also includes the original statistic value, T_0. Its p-value, as estimated from the EDF, can be read off as

$$p = 1 - \widetilde{F}(T_0) . \tag{12}$$

If the p-value is small then this is an indication that T_0 is in some sense unusual. We shall see how this idea can be developed into a full methodology for making different kinds of comparisons.

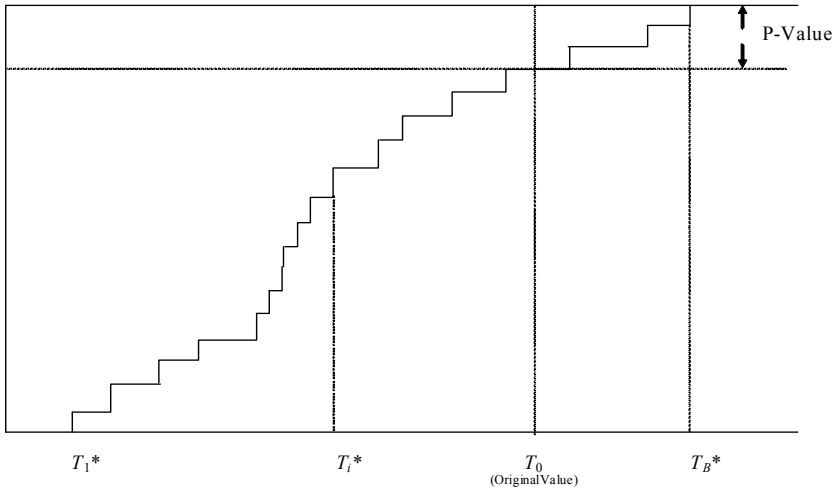


Figure 6: Empirical distribution function of the $T^{(i)}$

In practice typical values of B used for bootstrapping are 500 or 1000, and such a value is generally large enough. With current computing power resampling 1000 values is, to all intents and purposes, instantaneous.

Chernick (1999) shows how generally applicable is the bootstrap listing some 83 pages of carefully selected references! A good handbook for bootstrap methods is Davison and Hinkley (1997).

The bootstrap samples are trivially easy to obtain using the algorithm given below in Section 9. There is no problem in calculating the bootstrap statistics T_i* from the bootstrap samples $\mathbf{Y}_i$*, as the procedure for doing this must already have been available to calculate T from **Y**. All the remaining examples discussed in the tutorial are essentially an application of the bootstrapping process of Figure 5 (and its *parametric* variant, which we will be describing in Section 11), with whatever procedure is necessary inserted to calculate T from **Y**. The following is an elementary example.

Example 7: The spreadsheet contains a random sample of 50 observations from an unknown distribution for which the sample mean has been obtained. Obtain 90% confidence limits for the unknown true mean. This is not easily achieved using standard methods as the sample is not normally distributed. Instead we calculate 90% confidence limits using bootstrapping. This is done in `<Bootstrap Mean>`. □

9 BOOTSTRAP EVALUATION OF MAXIMUM LIKELIHOOD ESTIMATORS

Let us apply the bootstrapping idea to the evaluation of the distribution of MLE's. All that is required is to treat the MLE, $\hat{\boldsymbol{\theta}}$ as being the statistic T of interest and to follow the scheme depicted in Figure 5. We generate bootstrap samples by resampling with replacement from the original sample. Then we use the code to produce the bootstrap T from this sample. The pseudocode for the entire bootstrap process is as follows:

```
// y = (y(1), y(2), ..., y(n)) is the original sample.
// T=T(y) is the calculation that produced T from y.
For k = 1 to B
{
    For i = 1 to n
    {
        j = Int[1 + n × Unif()]        // Unif() returns a uniformly distributed U(0,1) variate each time
                                       // it is called. Int[.] returns the integer part of its argument
        y*(i) = y(j)
    }
    T*(k) = T(y*)
}
```

The simplicity of bootstrapping is now apparent. The resampling is trivially easy. The step that produces $T^*(k)$ invokes the procedure that produced T from $\boldsymbol{y}$, only $\boldsymbol{y}$ is replaced by $\boldsymbol{y}^*$. The key point here is that no matter how elaborate the original procedure was to produce T, we will *already* have it available, as we must have set it up in order to calculate $T = T(\boldsymbol{y})$ in the first place. The bootstrap procedure simply calls it a further B times.

Example 8: Use bootstrapping to produce 100 bootstrap versions of $\hat{\alpha}$ and $\hat{\beta}$ of Example 5, and hence obtain bootstrap confidence intervals for α_0 and β_0. Compare these with the confidence intervals obtained for α_0 and β_0 using asymptotic normality theory. The bootstrapping can be carried out using the "Bootstrap" worksheet in `<Gamma Bootstrap>`.

Produce confidence intervals for the waiting time in the queue using bootstrapping. Again compare these results with those produced by asymptotic theory. Do this using the "PerformanceIndex" worksheet in `<Gamma Bootstrap>`. □

10 COMPARING SAMPLES USING THE BASIC BOOTSTRAP

Next we consider the problem where we have two samples of observations **Y** and **Z** and we wish to know if they have been drawn from the same or different distributions. We consider how the basic bootstrap provides a convenient way of answering this question.

To illustrate the ideas involved we consider the simple situation where we have calculated the same statistic from each of the samples **Y** and **Z.** This statistic might be the sample mean or sample variance say. We shall call this statistic S and denote its values calculated from the two samples by $S(\mathbf{Y})$ and $S(\mathbf{Z})$. An obvious statistic to use for the comparison, which we call the *comparator statistic*, is the difference $T = S(\mathbf{Y}) - S(\mathbf{Z})$. We therefore need the null distribution of T, corresponding to when **Y** and **Z** have been drawn from the *same* distribution. This null situation is depicted in Figure 7

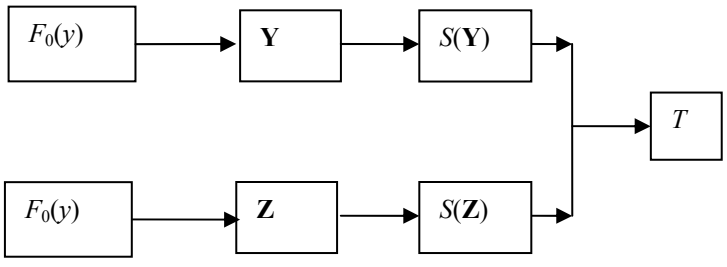


Figure 7: Calculation of a comparator statistic under the null hypothesis that both samples **Y** and **Z** are from the same distribution.

The procedure for producing a bootstrap version of T under the (null) assumption that the two samples are drawn from the same distribution is simple. We obtain bootstrap samples **Y*** and **Z*** from just one of the samples **Y** say. The bootstrap process is given in Figure 8. In the Figure $\widetilde{F}(y \mid \mathbf{Y})$ is the EDF constructed from **Y.** Alternatively the EDF $\widetilde{F}(y \mid \mathbf{Z})$ of **Z,** or perhaps even better, the EDF $\widetilde{F}(y \mid \mathbf{Y}, \mathbf{Z})$ of the two samples **Y** and **Z** when they are combined, could be used.

The test follows the standard test procedure. We calculate the *p*–value of the original comparator statistic T relative to the bootstrap EDF of the $\{T_i^*\}$. The null hypothesis that the two samples **Y** and **Z** are drawn from the same distribution is then rejected if the *p*–value is too small.

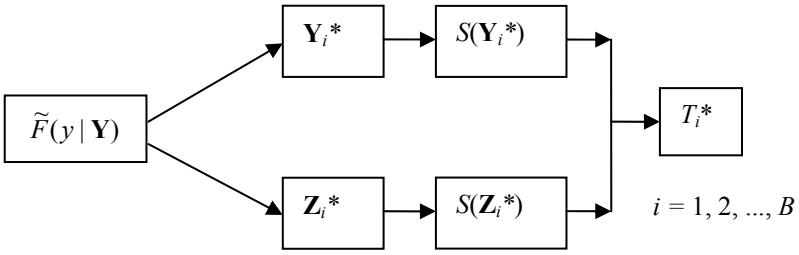


Figure 8: Bootstrap comparison of two samples under the null hypothesis

Example 9: Consider the data given in Law and Kelton (1991) recording the results of a simulation experiment comparing the costs of five different inventory policies. Five simulation runs were made for each of the five policies. The cost of running the inventory was recorded in each run, together with the means obtained using each policy. Use bootstrapping to see if there are any significant differences between the five means. Results and analysis are in <Law and Kelton EG>. □

11 THE PARAMETRIC BOOTSTRAP

The previous discussion has considered the basic bootstrap. There is a second method of bootstrapping termed the parametric bootstrap.

Suppose we have fitted a parametric model to data. If the parametric model is the correct one and describes the form of the data accurately, then the fitted parametric model will be a close representation of the unknown true parametric model. We can therefore generate bootstrap samples not by resampling from the original data, but by sampling from the fitted parametric model. This is called the *parametric bootstrap*.

The basic process is depicted in Figure 9. It will be seen that the method is identical in structure to the basic bootstrap. (Compare Figure 9 with Figure 5.) The only difference is that in the parametric version samples are generated from a given parametric distribution, e.g. the gamma distribution. A method, such as the inverse transform method, has to be available for doing this.

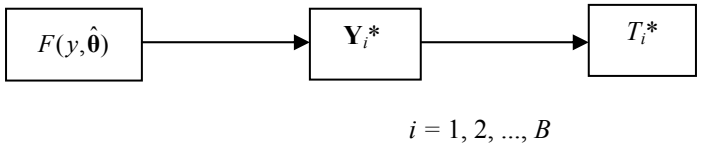


Figure 9: Parametric bootstrap process

Example 10: Consider the queueing example where we have already used asymptotic theory (Example 6) and the basic bootstrap (Example 8) to analyze the effect of estimated parameter uncertainty. Repeat the exercise but now use the parametric bootstrap instead of the basic bootstrap. Go to the worksheet "Bootstrap" in <ParametricBS-GammaEG> to do this.
□

At first sight the parametric bootstrap does not seem to be a particularly good idea because it adds an extra layer of uncertainty into the process, requiring selection of a model that may or may not be right. However there are situations where its use is advantageous and we discuss one in the next section.

12 GOODNESS OF FIT TESTING

12.1 Classical Goodness of Fit

We consider the natural question: *Does the model that we have fitted actually fit the data very well?* The classical way to answer this question is to use a *goodness of fit test* (GOF test). A very popular test is the *chi-squared goodness of fit test*. The main reason for its popularity is that it is relatively easy to implement, but it is not a very sensitive test to use.

The best general GOF tests directly compare the EDF with the fitted CDF. Such tests are called *EDF goodness of fit tests*. The best is probably the *Anderson - Darling* test, and this is the method of choice. The trouble with these tests is that, because of their sensitivity, their critical values are very dependent on the model being tested, and on whether the model has been fitted (with parameters having to be estimated in consequence). This means that different tables of test values are required for different models (see d'Agostino and Stephens, 1986).

First we describe EDF tests in more detail. Typically an *EDF test statistic* has the form

$$T = \int \psi(y)\left(\widetilde{F}(y) - F_0(y)\right)^2 dF_0(y) .$$

Here $\psi(y)$ is a weighting function. The *Anderson-Darling test statistic* has $\psi(y) = [F_0(y)(1 - F_0(x))]^{-1}$. Computationally this is equivalent to

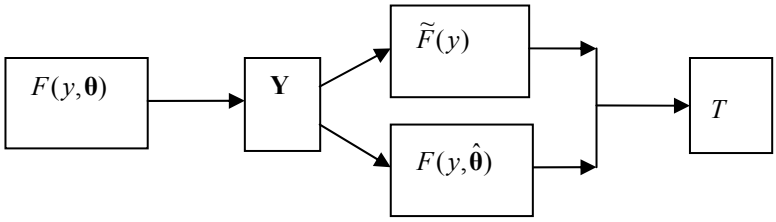
$$T = A^2 = -\sum_{i=1}^{n}(2i-1)[\ln Z_i + \ln(1 - Z_{n+1-i})]/n) - n$$

where $Z_i = F(Y_{(i)}, \hat{\boldsymbol{\theta}})$ is the value of the fitted CDF at the ith ordered observation.

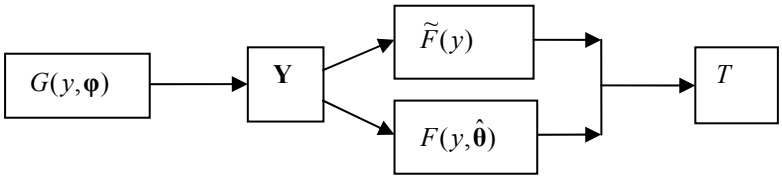
The basic idea in using a goodness of fit test statistic is as follows. When the sample has really been drawn from $F_0(y)$ then the value of the test statistic will be small. This follows from the Fundamental Theorem of Section 7 which guarantees that $\widetilde{F}(y)$, the EDF of the sample, will be close in value to $F_0(y)$ across the range of possible y values. Thus T will be small. Nevertheless because the test statistic is a random quantity, it will have some variability according to a *null distribution* depending on the sample size n. If the null distribution is known then we can assess an observed value of T against this distribution. If the sample is drawn from a distribution different from $F_0(y)$ then the T will be large. Statistically, what is conventionally called its *p - value* will then be small, indicating that the distribution has *not* been drawn from the supposed null distribution.

Figure 10 illustrates the process involved in calculating a GOF test statistic for the parametric case. Two cases are shown: in the first case, the correct model $F(y,\boldsymbol{\theta})$ is chosen for the parametric bootstrapping and in the second case, the true model $G(y,\boldsymbol{\varphi})$ differs from the model used. In both cases the distribution from which $\mathbf{Y}$ has been drawn is assumed to be $F(y,\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is unknown; thus in each case $F(y,\boldsymbol{\theta})$ has been fitted to the random sample $\mathbf{Y}$ giving the ML estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$. However in the first case $F(y,\boldsymbol{\theta})$ is the correct model. Thus $\widetilde{F}(y)$ will be the EDF of a sample drawn from $F(y,\boldsymbol{\theta})$ which will therefore converge to this distribution. In the second case the true model, $G(y,\boldsymbol{\varphi})$, is different from $F(y,\boldsymbol{\theta})$, and may even involve a set of parameters $\boldsymbol{\varphi}$ that is different from $\boldsymbol{\theta}$. Thus the EDF $\widetilde{F}(y)$ in this second case will *not* converge to $F(y,\boldsymbol{\theta})$, but to $G(y,\boldsymbol{\varphi})$. Thus in the second case, T, which is a measure of the difference between the two, will be larger than in the first case.

The *null situation* is the first case where we are fitting the correct model. We need to calculate the distribution of T for this case. A complication arises because the difference between $\widetilde{F}(y)$ and $F(y,\hat{\boldsymbol{\theta}})$ is *smaller* than the difference between $\widetilde{F}(y)$ and the unknown true $F(y,\boldsymbol{\theta})$. This is because the fitted distribution $F(y,\hat{\boldsymbol{\theta}})$ will follow the sample more closely than $F(y,\boldsymbol{\theta})$ because it has been fitted to the sample. This has to be allowed for in calculating the null distribution of the test statistic.



Null Case: Fitted model $F(y,\hat{\boldsymbol{\theta}})$ is the correct model.



Alternative Case: Fitted model, $F(y,\hat{\boldsymbol{\theta}})$, is an incorrect model.

Figure 10: Process underlying the calculation of a GOF test, T

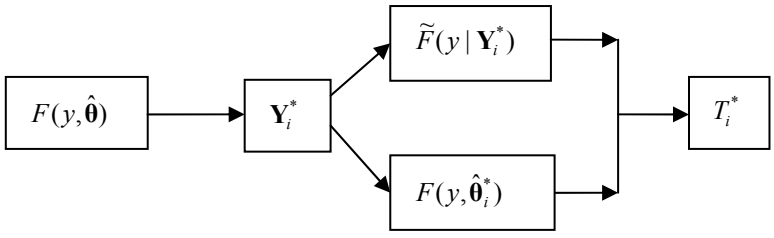
It will be seen that the GOF test hinges on being able to calculate the null distribution. This is a big issue and has meant that many potentially powerful test statistics, like the Cramér - von Mises, have not been fully utilized in practice because the null distribution is difficult to obtain. In the next subsection we show how resampling provides a simple and accurate way of resolving this problem.

12.2 Bootstrapping a GOF statistic

The null case calculation of the GOF statistic depicted in Figure 10 is identical to that of Figure 4. Thus if we could obtain many values of T using this process then the EDF of the T_i will converge to the CDF of T. This is almost certainly too expensive or impractical to do. However we can get a close approximation by simply replacing the unknown $\boldsymbol{\theta}$ by its MLE $\hat{\boldsymbol{\theta}}$. This is precisely the parametric bootstrap process as given in Figure 9.

All being well $\hat{\boldsymbol{\theta}}$ will be close in value to $\boldsymbol{\theta}$. Thus the distributional properties of the T_i^* will be close to those of a set of T_i obtained under the null case calculation of Figure 10.

The parametric bootstrap method as it applies to a GOF statistic is illustrated in more detail in Figure 11, where $\widetilde{F}(y \mid \mathbf{Y}_i^*)$ is the EDF of the bootstrap sample $\mathbf{Y}_i^*$, and $\hat{\boldsymbol{\theta}}_i^*$ is the bootstrap MLE of $\hat{\boldsymbol{\theta}}$ obtained from the bootstrap sample $\mathbf{Y}_i^*$



$i = 1,2, ..., B$

Figure 11: Bootstrap process to calculate the distribution of a GOF test, T

Example 11: Examine, using the bootstrapping, whether the gamma model is a good fit to the toll booth data. Examine also whether the normal model is a good fit to the toll booth data. Use the Anderson-Darling goodness of fit statistic A^2, previously given. Macros called from the worksheet "Bootstrap" in each of the following workbooks, will carry out this analysis.
<Gamma Fit Toll Booth Data> <Normal Fit Toll Booth Data> □

13 COMPARISON OF DIFFERENT MODELS: MODEL SELECTION

We consider the situation where we have a number of alternative models that might be fitted to a set of data, and we wish to choose which model is the best fit. We shall only discuss the multiple linear regression case where Y depends on a number of factors

$$Y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_M X_M + \varepsilon_i, \quad i = 1,2,...,n$$

but it is not clear which factors are important. Here a decision has to be taken on each factor as to whether to include it or not. A good criterion to select between models is Mallow's Statistic (see Mallows, 1973, 1995)

$$C_k = \frac{RSS_k}{RMS_{M+1}} + 2k - n,$$

where RSS_k is the regression sum of squares from fitting k - 1 factors and RMS_{M+1} is the residual mean square from fitting the full model. Assuming that the constant β_0 is always fitted there are a total of 2^M different possible models in all so that a systematic search through all possible models will be computationally expensive. However for the case where the experimental design used to estimate the parameters is orthogonal then use of Mallow's statistic is equivalent to looking at each parameter estimate separately and to include factor j in the model if $t_j = \hat{\beta}_j / SE(\hat{\beta}_j) > \sqrt{2}$ where $SE(\hat{\beta}_j)$ is the standard error of the estimator $\hat{\beta}_j$. Once a 'best' model has been chosen in this way, what degree of confidence can we place on the choice made? Bootstrapping provides a very effective way of answering this question.

Example 12 (continuation of Example 2): In the Ericsson supply chain model which of the 16 factors are important? This is investigated using bootstrapping in <Ericsson EG>. □

This completes our discussion of some of the basic problems that regularly occur in modeling.

14 FINAL COMMENTS

This tutorial has reviewed the process of constructing and fitting a statistical model to data whether this data arises from study of a real system or from a simulation.

The classical approach using the method of maximum likelihood has been described for fitting the model.

Two approaches for assessing the variability of fitted parameters have been discussed: (i) The classical approach using asymptotic normality theory and (ii) Bootstrap resampling.

Examples have been drawn mainly from regression and ANOVA applications. These have been for illustration only. We have not attempted to survey the range of statistical models likely to be encountered in practice, where typically different aspects of modeling need to be brought together. Krzanowski (1998) gives a very comprehensive but at the same time very accessible survey of the different types of situation commonly encountered. For instance, Krzanowski's Example 6.1 gives data relating to factors affecting the risk of heart problems: social class, smoking, alcohol and so on. The data is 'binary response' data (i.e. a patient reporting some form of 'heart trouble' or not). The factors are *categorical* (for example alcohol is coded as someone who drinks or someone who does not). The required model is therefore a logistic regression model but with a linear predictor involving the categorical factors. Though this example has not been discussed explicitly in this tutorial, all the elements needed to analyze it using either classical or bootstrap methods have been considered, and despite its apparent complexity, this model is quite capable of being tackled straightforwardly using the methods of this tutorial.

In fact the spreadsheets given for the examples in this tutorial use a number of VBA macros that enable various commonly occurring analyses to be carried out. These macros have been designed to be sufficiently flexible and accessible to be used in other applications and you are encouraged to make use of them in this way. To assist in this, one last example is included.

Example 13: This shows how a regression model can be fitted to the traffic queue data of Example 1. The model allows the 'error' variability to depend on the traffic intensity. The workbook <RegressionFitTrafficQueueData> is set out much like the main workbook used in the tutorial for analysing the Toll booth data of Example 3. It includes both a classical asymptotic and a bootstrap analysis of the accuracy of the fitted regression line. A guide to how the spreadsheet is provided in <GuideToRegressionFitTrafficEG>. □

REFERENCES

Chernick, M. R. 1999. Bootstrap Methods, A Practitioner's Guide. New York: Wiley.
D'Agostino, R.B. and M.A. Stephens 1986. Goodness of Fit Techniques. New York: Marcel Dekker.
Davison, A.C., and D. V. Hinkley. 1997. Bootstrap Methods and their Application. Cambridge: Cambridge University Press.
Kleijnen, J.P.C., B. Bettonvil, and F. Persson. 2006. Screening for the Important Factors in Large Discrete-Event Simulation Models: Sequential Bifurcation and Its Applications. Chapter13, in Screening, Methods for Experimentation in Industry, Drug Discovery, and Genetics. Eds A. Dean and S Lewis. Springer.
Krzanowski, W. J. 1998. An Introduction to Statistical Modelling. London: Arnold.
Law, A.M., and W. D. Kelton. 1991. Simulation Modeling and analysis, 2nd Ed., New York: McGraw-Hill.
Mallows, C. L. 1973. Some comments on C_p. Technometrics, 15, 661-675.
Mallows, C. L. 1995. More comments on C_p. Technometrics 37, 362-372.
Urban Hjorth, J.S. 1994. Computer Intensive Statistical Methods. London: Chapman & Hall.

AUTHOR BIOGRAPHIES

RUSSELL C. H. CHENG is Emeritus Professor of Operational Research at the University of Southampton. He has an M.A. and the Diploma in Mathematical Statistics from Cambridge University, England. He obtained his Ph.D. from Bath University. He is a former Chairman of the U.K. Simulation Society, a Fellow of the Royal Statistical Society and a Member of the Operational Research Society. His research interests include: design and analysis of simulation experiments and parametric estimation methods. He was a Joint Editor of the *IMA Journal of Management Mathematics*. His email and web addresses are <R.C.H.Cheng@soton.ac.uk> and <www.personal.soton.ac.uk/rchc>.

CHRISTINE S. M. CURRIE is a lecturer of Operational Research in the School of Mathematics in the University of Southampton, where she also obtained her Ph.D. In addition she has an MPhys from Oxford University and an MSc in Operational Research from the University of Southampton. She is currently the Book Review Editor for the Journal of Simulation and co-chair of the Simulation Special Interest Group in the UK Operational Research Society. Her research interests include mathematical modeling of epidemics, Bayesian statistics, revenue management, variance reduction methods and optimization of simulation models. Her email address is <christine.currie@soton.ac.uk> and her web page is <www.personal.soton.ac.uk/ccurrie>.