**NATCOR Simulation**

**————————————————————————————**

**Topic: Input Modelling, Simulation Optimization and Design and Analysis of Simulation Experiments**

The lectures are based on *Working Notes* which are divided into four Parts.

You can access the working notes by going to the bottom of Russell Cheng’s Home Page and following the link there. (To access Russell’s Home Page just type his name into Google and follow the link to his Home Page.)

The Working Notes are meant to be *worked through*.

They contain *Examples* and *Exercises*. These illustrate the topic or method being discussed. They are an essential part of the text and need to be carefully studied.

Many of the Examples and Exercises come with their own link. (i) Some of the links contain additional notes and more detailed formulas, (ii) The other links are to actual spreadsheets containing data and the worked details using the data.

Some of the initial spreadsheets contain elementary exercises connected with generating random variables and simple sampling experiments. You should aim to do these exercises yourself independently of the worked solutions and then compare your solution with that supplied. The point of these exercises is to give you familiarity with basic formulas and functions that you will need for the more complicated later examples.

The other spreadsheets contain more substantial problems. These are solved using VBA macros for carrying out more substantial calculations and more extensive analyses. You are ***not*** expected to write your own macros to duplicate these macros. However you should spend sufficient time using and studying the macros to understand how they function. ***Thus you should aim to be able to understand the workings of the VBA macros sufficiently well to be able to modify them for solving simple variations of the problem to which they have presently been applied.*** I have tried to make the macros transparent and relatively easy to modify.

In the spreadsheets, the following convention for cells is used:

Cells with a Yellow background - Headings, Incidental Information

Cells with a Green background - Input Information used in calculations on that Sheet

Intermediate Results and Calculations are not usually coloured.

**————————————————————————————**

Links

• [Working Notes: Part I](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\CADMPartI.htm)

• [Working Notes: Part II](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\CADMPartII.htm)

• [Working Notes: Part III](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\CADMPartIII.htm)

• [Working Notes: Part IV](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\CADMPartIII.htm)

• [Lectures/Labs Diary](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\CADMContentsDiary.htm)

Lab WorkSheets

[Worksheet 1](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\Worksheet1ExamineTollBoothSpreadsheet.doc)

[Worksheet 2](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\Worksheet2FitModelToTrafficQueueData.doc)

[Worksheet 3](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\WorkSheet3FitModelToCortisolData.doc)

[Worksheet 4](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\Worksheet4TyreANOVAanalysis.doc)

References are provided at the end of Part IV

**NATCOR Simulation**

**Topic of Study: Input Modelling, Simulation Optimization and Design and Analysis of Simulation Experiments**

**Contents**

**I Statistical Metamodels**

1. Introduction

2. Statistical MetaModels

**II Input Modelling**

3. Introduction

4. Random Variables

5. Fitting Parametric Distributions to Random Samples

6. Maximum Likelihood Estimation

7. Accuracy of MLEs

8. Goodness of Fit Testing

9. Extension: bootstrapping the Goodness of Fit statistic

**III Simulation Optimisation**

10. Introduction

11. Ranking and Selection

12. Stochastic Approximation

13. Random Search

**IV Experimental Design and Analysis of Linear Models**

14. Linear Regression Metamodels

15. Fitting and Assessing the Linear Model

15.1 Least Squares Estimation

15.2 ANOVA

15.3 Individual Coefficients

16. Prediction with the Linear Model

17. Additional Explanatory Variables

18. Experimental Designs

18.1 Main Effects Model

18.2 Factorial Designs

18.3 Plackett-Burman Designs

19. Interactions

20. Central Composite Designs

21. Comments on Design of Experiments

22. Final Comments

**References**

**NATCOR Simulation**

**Topic: Experimental Design and Analysis**

**Part I Statistical MetaModels**

**1. Introduction**

The topic of experimental design and analysis is a general one, though in this course, it is being presented in the context of simulation studies. As will become plain in the next section, when we discuss metamodels, the core of the topic is statistical methodology as it applies, not simply just to simulation models, but to *all mathematical modelling* where systems are being studied that are subject to random perturbations. You should therefore find the topic of much more general use than simply in simulation modelling studies, and I encourage you to adopt the approach that I will be using not only in simulation modelling but in the study of any system where random behaviour has to be accounted for.

The material that we shall study covers what I have found most useful in tackling such a modelling problem - the sort of things that in retrospect I wished had been pointed out to me when I first encountered problems of this sort.

If you have covered intermediate Statistics or OR, you may already have encountered most of the methods to be discussed. However these will usually have been taught somewhat in isolation, each within a formal, even disembodied, setting in which there is not enough scope, or time, to emphasize the overall way that these methods invariably come together when studying a simulation or a modelling problem. There is a natural way in which the methods are used, giving them a power and unity that is not always apparent when they are first encountered.

I shall try and emphasise what should be going through the mind of the investigator at each stage, showing that an overall problem can be broken down into a standard set of subproblems all of which will invariably occur, and in consequence showing that they can be tackled in a unified way.

By the end of this part of the course the student should be much more assured in the way that she/he confronts and tackles such a modelling exercise. There will be a much better awareness of the steps needed to carry out an exercise successfully, and of the problems and issues that occur at each step.

A very good book that has a similar philosophy to this course 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.

**2. Statistical MetaModels**.

This course emphasizes 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 2.1 illustrates the situation where we have data, **Y** (here and throughout this text, a quantityis written in bold to indicate that it is a vector quantity), available concerning the behaviour.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 analyse, to learn about the behaviour 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 **X** = (*X*1, *X*2, … , *Xk*) is a vector of *k* explanatory variables. These are known quantities and indeed may possibly be under the control of the investigator. The input **θ** = (*θ*1, *θ*2, …, *θp*) 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*.

In addition the output **Y** may contain a random component, typically referred to as ‘noise’ or ‘error’. This is denoted by **ε**.

Figure 2.1: Schematic of the System

Input

**X**

1: Real System

2: Simulation model

3: Statistical metamodel

**Y**

Input



Output

Data

Noise 

As well as depicting the situation where the output data has been obtained from a real system Figure 2.1 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 2.1 by replacing the real system in the central block by a simulation model. All other blocks remain the same.

In this course the focus is on how to analyse **Y** and in particular to identify how the inputs **X** and  influence **Y** in the presence of the random effects . 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 2.1 yet again, only with the central block now representing the statistical model.

The term statistical model is conventionally used when we are analysing 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.

**Example 2.1:** Consider the operation of a queue where we are interested in estimating, the average queue length, over a given period of time, *T* say. 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, and *C* might be the number of servers available. [Traffic Queue Length EG](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\TrafficQueueEG.xls)

This is a situation that has been well analysed theoretically and where the relationship between *Y* and the quantities *λ*, *μ* and *C* is known precisely in certain situations. However we might be uncertain about the precise form of inter-arrival and service time distributions. We can assume that the results of *n* simulation runs take the form

 (2.1)

where  is some suitably selected function characterising the likely behaviour of *Y*. The quantity  is a random variable. A common assumption is that the errors have a normal distribution:

~ . (2.2)

This assumption is questionable in the present context as the variability of *Y* will depend critically on the traffic intensity , so the assumption of constant variance  for the observations *Y*  is dubious. □

**Example 2.2:** The National Health Service has data for, *Y*, the number of newly registered diabetics in each year for a given number of years. It also has data on a selection of factors that might influence the onset of diabetes such as, *X*1 amount of alcohol consumed; *X*2the number of cigarettes smoked, per day; X3; previous illnesses contracted, age, sex of each case. The problem here is to those identify factors that have a significant influence on the onset of diabetes. A typical model is

. (2.3)

where *yi* is the observed number of registered diabetics in year *i*  and *xij* is the observed value of the *j*th factor in year *i*; and we have *n* years of observations. Again we might assume normal errors  ~ . □

The scenario of Figure 2.1 can be varied or extended in many different ways. We illustrate this with two commonly occurring situations.

The first is illustrated in Figure 2.2. This is the situation where the input **θ** parameters can be *estimated* using data or past information, **w**, containing information about **θ**. Sometimes this information is not explicit but is derived from expert opinion. The estimation in this latter case is then possibly subjective. We write these estimates as , or simply as  depending on whether past data **w** is involved or not, using the circumflex to indicate an estimated quantity.

Figure 2.2: Input Parameters Estimated from Data

Input

X

System/

Model

Output



Input



Data

**w**

Noise

**ε**

**Example 2.1 (continued):** It may be that *λ*, the arrival rate, is not known. However we have a sample of interarrival times from which *λ* can be estimated. □

Another important variation is when a dynamic system or simulation is being examined. Here *time* - which we denote by *t* - enters into the picture. It is usually best to treat *t* as being a continuously varying input variable that is part of the input **X**, but which then results in the output *Y* being time-dependent. Figure 2.3 represents this time dependent situation.

Figure 2.3: Schematic of a Dynamic System/Model

Data

**w**

System/

Model

Output





Noise

**ε**

Input



Input

X

**Example 2.3:** In the study of an epidemic let *η*(*t*, **θ**) represent the prevalence of a certain disease at time *t*. (Prevalence means the proportion of the population who has the given disease.) Several scenarios are possible. Firstly, it may be that there is no information on **θ**, but there are observations *yi* of the prevalence at given time points *ti* *i =*1, 2, ..., *n*. These are subject to error, thus

*yi* = *η*(*ti,* **θ**) + *εi*, *i =*1, 2, ..., *n*. (2.4)

Then the problem would be to fit **θ** to the observations {*yi*}. Secondly we might have past information **w** on which **θ** depends and the task is then to estimate **θ** from the information **w.** The third possibility is when we have both observations {*yi*} of the epidemic and there is past information **w** on the parameters **θ**. In this case we should use both the {*yi*} and **w** to estimate **θ**.

The following is a set of data giving the number of notifications of pulmonary TB (per 100,000) in Morocco in four selected years 1980, 1986, 1993, 2000, grouped by age. What form should *η*(*t*, **θ**) take? [Moroccan TB Data](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\MoroccanTBData.xls) □

In dynamic problems the regression formulation (4) is typical. The regression function *η*(*t*, **θ**) has to be selected so that its behaviour resembles the output of the system or mathematical/simulation model that it represents. In some situations, as might occur in the dynamic case just considered, the physical process of the actual system may be sufficiently known to suggest a natural form for *η*(*t*, **θ**).

**Example 2.4:** The logistical curve

 (2.5)

is commonly used to represent population growth when this takes a sigmoidal form. □

**Exercise 2.1:** Plot the logistic curve on a spreadsheet for different combinations of *α*, *β*, *γ*. □

If little is known about the real system, the form assumed for *η*(*t*, **θ**) 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:

 (2.6)

When there are a large number of factors, and especially when the errors **ε** are not small then a multivariate linear form is often used:

 (2.3 bis)

Here the *xi* are the values of the different factors, and the model only considers the inclusion of a linear term for each factor. Example 2.2 is an illustration of a situation where this multivariate form is appropriate.

Sometimes the output *Y* takes a binary form, indicating success (*Y =* 1) or failure (*Y* = 0). Representing *Y* in terms of a continuous function is not then very sensible. The usual ploy is to model the probability *π* that *Y* = 1 and then to ensure that *π* lies between 0 and 1 by using a transformation such as the logistic transformation. This is usually written as

 (2.7)

but the more correct version is

 (2.8)

as (in the one *x* variable case) it is actually

 (2.9)

that is the logistic transform.

This binary model is best not thought of in regression terms. Instead we regard each observation as a Bernoulli variable

*Yi* ~ *Bernoulli*( ) (2.10)

**Example 2.5:** An example of binary response data:[Vaso Constriction Data](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\VasoConstriction.xls)

As far as this course is concerned we will be focusing on the third representation of Figure 2.1 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 explicit how the distribution is expected to depend on the input quantities. It should be stressed that the statistical model does not have to precisely copy the characteristics of the underlying true model, which anyway may be too complicated to be sensibly reproducible. Rather the statistical model has to be capable of modelling the essential features of the system it represents, but that is all that is needed. Figure 2.4 illustrates this key requirement, in the dynamic case, by including boxes to represent both the unknown system and the statistical model representing it. The parameters  of the statistical model do not have to correspond in any explicit way to the parameters **θ** of the system, and this is indicated in Figure 2.4

Often the regression format is a convenient one to use. However, as the last example shows, the regression approach is not completely general. In fact the procedure used in equation (2.8) of the last example, of treating *Y* as a random variable and writing down its distributionby *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.

Figure 2.4: This depicts a Statistical Model of a System.

It also depicts a Metamodel of a Simulation Model

Statistical Model..

Output



Output



Input

X

System/

Model

Input



Data

**w**



Noise

**U**

The main conclusion to draw from our consideration of statistical metamodels is summarized in the following :

***The 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 Part II.

**Part II Input Modelling**

**3. Introduction**

Input modelling for simulation models involves finding the best way of specifying inputs to the simulation model. These might be the distribution of service times in a queueing system; the arrival process into the simulated system; the process of modelling machine breakdowns in a manufacturing simulation; etc. A model is only as good as the data you use to build it or, as the saying goes, garbage in – garbage out. Consequently determining the correct input models and quantifying the uncertainty associated with the choice of input model is important.

In this section, we focus on situations in which data are available to fit input models and we use parametric distributions to describe the distribution of the data. Before describing the fitting process, we will discuss some useful ideas from mathematical statistics.

**4. Random Variables**

The key concept of all statistics is the *random variable*. A formal definition of a random variable requires a mathematical foundation (and elaboration) that takes us away from the main focus of this course. We shall therefore not attempt a formal definition but instead adopt a simpler practical viewpoint. We therefore define 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 thus 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).

In view of the above discussion, given a random variable, one should immediately think of the *range of possible values* that it can take and its *probability distribution* over this range.

The definition of most statistical probability distributions involves *parameters*. Such a probability distribution is completely fixed once the parameter values are known. Well known parametric probability distributions are the normal, exponential, gamma, binomial and Poisson. The next section provides a brief description of the best-known parametric probability distributions.

A probability distribution is usually either *discrete* or *continuous*. A discrete distribution takes a specific set of values, typically the integers 0, 1, 2,…. Each value *i* has a given probability *pi* of occurring. This set of probabilities is called its *probability mass function*.

**Exercise 4.1:** Plot the probability mass function of

1. the binomial distribution, B(*n*, *p*)
2. the Poisson distribution, P(*λ*)

Write down what you know about each distribution. □

A continuous random variable, as its name implies, takes a continuous range of values for example all *y ≥* 0. One way of defining its distribution is to give its *probability density function* (pdf), typically written as *f*(*y*). The pdf is *not* a probability, however it can be used to form a *probability increment*.  This is a good way to view the pdf.

**Exercise 4.2:** Write down the pdf of

(i) the normal distribution, . [Normal Distribution](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\ExtraDetails\Normal%20Distribution.doc)

(ii) the gamma distribution, . [Gamma Distribution](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\ExtraDetails\Gamma%20Distribution.doc)

Plot the density functions. Write down what you know about each distribution. □

**Exercise 4.3:** Suppose that *X* is a continuous random variable with density *f*(*x*). Let *Y* be a function of *X*, say *Y* = *h*(*X*). What is the pdf, *g*(*y*) of *Y*, in terms of *f*(*x*)? Give the pdf of *Y* = *X* 2 when *X* is a standard normal random variable. What is the name of this random variable and what is the form of its pdf? □

An alternative way of defining a probability distribution, which applies to either a discrete or continuous distribution, is to give its *cumulative distribution function* (cdf).

**Exercise 4.4**: Write down the main properties of a cdf. □

**Exercise 4.5**: Plot the cdf’s of each of the examples in the previous examples. □

**Exercise 4.6**: What is the relation between the pdf and the cdf for a continuous random variable? How is one obtained from the other? □

**Exercise 4.7**: Define the expected value of a random variable *X* in terms of its pdf *f*(*x*). Define the *expected value* of *Y* = *h*(*X*) in terms of the pdf of *X*. □

**4.1. Empirical Distribution Functions**

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

. (4.1)

The EDF is illustrated in Figure 4.1. It is usually simplest to think of the observations as being *ordered*:

*Y*(1) < *Y*(2) < … < *Y*(*n*) . (4.2)

These are what are depicted in Figure 4.1. Note that the subscripts are placed in brackets to indicate that this is an ordered sample.

The key point is that the EDF estimates the (unknown) *cumulative distribution function* (CDF) of *Y*.

**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 rigour, 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.)

Figure 4.1: EDF of the *Yi* , 

*Y*(*n*)

*Y*(*j*)

*Y*(1)

We can study the properties of the *Yi*  directly using the EDF, without bothering to fit a parametric model. If the *Yi* describe the input of a simulation model, this can be viewed as a way of ensuring that we do not generate values that could not exist in the real system. However, there are drawbacks such as the incomplete coverage of the range of feasible values and the inability to sample values outside of the range of observations.

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*. Thus *Y* can be either a continuous or a discrete random variable. Nor is it assumed to come from any particular family of distributions like the normal or Weibull. This flexibility is particularly important when the EDF is that of the output of a complex simulation, where it is possible that the distribution of the output may be unusual. For example it may well be skew, or possibly even multimodal.

**5. Fitting Parametric Distributions to Random Samples**

*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 **Y** = {*Y*1, *Y*2, … *Yn*,} where each *Yi* represents one of the observations.

Parametric distributions are statistical distributions that we can write down mathematical expressions for that involve parameters. These parameters need to be estimated and this is the main focus of this and the next section. Before considering the fitting, it is worth reminding ourselves of the main characteristics of some of the most popular statistical distributions and these are given in Tables 5.1 (discrete distributions) and 5.2 (continuous distributions)

Table 5.1 Common Discrete Probability Distributions

|  |  |  |
| --- | --- | --- |
| Name | Description | Probability Mass Function |
| Poisson | Probability of a given number of events *k* occurring in a fixed interval of time or space. |  |
| Binomial | Probability of a number of successes *k*, in *n* independent yes/no experiments. |  |
| Geometric | Probability of the number of failures *k*, before the first success in a sequence of Bernoulli trials. |  |

Table 5.2 Common Continuous Probability Distributions

| Name | Description | Probability Density Function |
| --- | --- | --- |
| Beta | Used to model the distribution of quantities that are bounded between 0 and 1 (although adding a constant or multiplying by a constant can change the range). | , *x*∈(0,1) |
| Exponential | Useful for modelling the time between independent events, e.g. inter-arrival times. If inter-arrival times follow an exponential distribution, the number of arrivals in a given time follows a Poisson distribution. |  |
| Gamma | Used to model non-negative random variables (the exponential is a special case of the gamma distribution) |  |
| Lognormal | Often used to describe the distribution of a quantity that is the product of a number of different quantities. |  |
| Normal | Often used to describe the distribution of a quantity that is the sum of a number of different quantities, e.g. the sum of times required to complete an operation. |  |
| Triangular | Often used when no data are available but the minimum, maximum and most likely values are known. |  |
| Uniform | Used when there is an equal probability of the quantity taking on any value in a given range. |  |
| Weibull | Most often used to model the time to failure. The exponential is a special case of the Weibull. |  |

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

**Example 2.1 (continued)**: Suppose we are modelling a queueing system where service times are expected to have a gamma distribution and we have some actual data of the service times of a number of customers from which we wish to estimate the parameters of the distribution. This is an example of the input modelling problem. If we can estimate the parameters of the distribution, we will have identified the 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, which as we shall see shortly possess a number of very desirable properties, are called *maximum likelihood estimates* (MLEs). ML estimation is a completely general method that applies not only to input modelling problems but to all parametric estimation problems. We describe the method next.

**6. Maximum Likelihood Estimation**

Suppose **Y** = {*Y*1, *Y*2, …, *Yn*} is a set of observations where the *i*th observation, *Yi*, is a random variable drawn from the continuous distribution with pdf *fi*(*y*, **θ**) (*i* = 1, 2, …, *n*). The subscript *i* indicates that the distributions of the *yi* can all be different.

**Example 6.1:** Suppose *Yi* ~ *N*(*μ*, *σ* 2) all *i*. In this case

** (5.1)

so that the observations are identically distributed. The set of observations is therefore a random sample in this case. □

**Example 6.2:** Suppose *Yi* ~ **, *i =* 1, 2, ..., *n*. In this case one often writes

 (5.2)

where

 (5.3)

is called the *regression function*, and

~ *N*(0, *σ*2) (5.4)

can be thought of as an *error* term, or a *perturbation* affecting proper observation of the regression function. □

In the example, the regression function is linear in both the parameters , and in the explanatory variable *x*. In general the regression function can be highly nonlinear in both the parameters and in the explanatory variables.

In Example 6.2, the pdf of *Yi* is

**. (5.5)

Thus **Y** is *not* a random sample in this case, because the observations are not all identically distributed. However ML estimation still works in this case.

We now describe the method. Suppose that **y** = {*y*1, *y*2, …, *yn*} is a sampled value of **Y** = {*Y*1, *Y*2, …, *Yn*}. Then we write down the joint distribution of **Y** evaluated at the sampled value **y** as:

. (5.6)

This expression, *treated as a function of* **θ,** is called the called the *likelihood* (of the sampled value **y**)**.** The logarithm:

 (5.7)

is called the *loglikelihood*.

*The ML estimate, , is that value of which maximizes the loglikelihood.*

The MLE is illustrated in Figure 6.1 in the one parameter case. In some cases the maximum can be obtained explicitly as the solution of the vector equation

 (15)

which identifies the stationary points of the likelihood. The maximum is often obtained at such a stationary point. This equation is called the *likelihood equation*. The MLE illustrated in Figure 6.1 corresponds to a stationary point.

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 6.1. The Maximum Likelihood Estimator 



Loglikelihood



There are two immediate and important points to realise in using the ML method.

(i) An expression for the likelihood needs to be written down using (6.6) or (6.7).

(ii) A method has to be available for carrying out the optimization.

We illustrate (i) with some examples.

**Exercise 6.1:** Write down the likelihood and loglikelihood for

(i) The sample of Example 6.1

(ii) The sample of Example 6.2

(iii) A sample of observations with the Bernoulli distributions (2.10).

(iv) The sample of Example 2.3

[Likelihood Examples](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\ExtraDetails\NATCORLikelihood%20Examples.doc) □

We now consider the second point, which concerns how to find the maximum of the likelihood. A number of powerful numerical optimizing methods exist 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. This is discussed in more detail in the following reference here: [Nelder Mead](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\ExtraDetails\NelderMeadProof.pdf).

**Exercise 6.2:** [NelderMeadDemo](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\NelderMeadDemo.xls) This is a VBA implementation of the Nelder-Mead Algorithm. Insert a function of your own to be optimized and see if it finds the optimum correctly.

Watchpoint: Check whether an optimizer minimizes or maximizes the objective. Nelder Mead usually does function minimization. □

**Exercise 6.3:** The following is a (random) sample of 47 observed times (in seconds) for vehicles to pay the toll at a booth when crossing the Severn River Bridge. Use the Nelder-Mead method to fit the gamma distribution *G*(*α*, *β*) to this data using the method of maximum likelihood. [Gamma MLE](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\GammaFitTollboothData.xls)

Watchpoints: Write down the loglikelihood for this example yourself, and check that you know how it is incorporated in the spreadsheet. □

**7. Accuracy of ML 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, **,is thatits asymptotic probability distribution is known to be normal. In fact it is known that, as the sample size *n* → ∞,

* ~ * (7.1)

where **is theunknown true parameter value and the variance has the form

**, (7.2)

where

** (7.3)

is called the *information matrix*. Thus the asymptotic variance of  is the inverse of the information matrix evaluated at . Its value cannot be computed precisely as it depends on the unknown **, but it can be approximated by

**. (7.4)

The expectation in the definition of ** is with respect to the joint distribution of **Y** and this expectation can be hard to evaluate. In practice the approximation

** '** (7.5)

where we replace the information matrix by its sample analogue, called the *observed information*, is quite adequate. Practical experience indicates that it tends to give a better indication of the actual variability of the MLE. Thus the working version of (7.1) is

* ~ *  (7.6)

The second derivative of the loglikelihood, **, that appears in the expression for ** is called the *Hessian* (of **L**). It 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 expectation, the expression ** can still be very messy to evaluate analytically. Again it is usually much easier to calculate this 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 (7.6) is typically used is to provide confidence intervals. For example a (1-α)100% confidence interval for the coefficient *θ*1 is

** (7.7)

where **is the upper 100α/2 percentage point of the standard normal distribution.

Often we are interested not in **θ** directly, but some arbitrary, but given function of **θ**, g(**θ**) say. ML estimation has the attractive general *invariant* property that the MLE of

g(**θ**) is

**. (7.8)

An approximate (1-α)100% confidence interval for g(**θ**) is then

** (7.9)

In this formula the first derivative of *g*(**θ**) 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.

Summarising 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 (Nelder-Mead say) numerical optimization of the loglikelihood.

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

(v) Calculate confidence intervals for these quantities.

**Example 7.1:** Suppose that the gamma distribution ** fitted to the toll booth data of Exercise 6.3 is used as the service distribution in the design of an M/G/1 queue. Suppose the interarrival time distribution is known to be exponential with pdf

 (7.10)

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

. (7.11)

Plot a graph of the mean waiting time  for the queue for 0 < *λ* < 0.1 (per second), assuming that the service time distribution is gamma: **. Add 95% confidence intervals to this graph to take into account the uncertainty concerning  because estimated values  have been used. [Gamma MLE](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\GammaFitTollboothData.xls) □

**Example 7.2:** Analyse the Moroccan Data of Example 2.3. Fit the model

 *i =* 1,2, .... 32

where *xi* is the age group of the *i*th observation,

,

and

 ~ *N*(0, *θ*12). [Regression Fit Morocco Data](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\RegressionFitMorocco.xls)  □

The previous two examples contains all the key steps in the fitting of a statistical model to data. Both examples involve regression situations. However the method extends easily to other situations like the Bernoulli model of Equation (2.10).

**Example 7.3:** Analyse the [Vaso Constriction Data](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\VasoConstrictionFit.xls) by fitting the Bernoulli model of Equation (2.10) using ML estimation. □

**8 Goodness of Fit Testing**

**8.1 Classical Goodness of Fit**

We consider the natural question: *Does the model that we have fitted actually fit the data very well?* For instance in Exercise 6.3 we fitted a gamma distribution to toll booth service time data, but does the fitted gamma distribution capture the characteristics of the data properly?

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. The test statistic is easy to calculate and moreover it has a *known* chi-squared distribution, under the null, which makes critical values easy to obtain.

However the chi-squared test has two obvious weaknesses. It is actually not all that powerful, and it has a certain subjective element because the user has to divide the data into groups of her/his own choosing.

The best general GOF tests directly compare the EDF with the fitted CDF. Such tests are called *EDF goodness of fit tests*. In the past the *Kolmogorov-Smirnov* test has been the most popular, but the *Cramér - von Mises* test and the *Anderson - Darling* test, defined below, are generally more powerful and should be used in preference. 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. Applying the Fundamental Theorem of Section 4, we see that a natural way to test if a sample has been drawn from the distribution with CDF *F*0(*y*), is to compare  with *F*0(*y*) by looking at some quantity involving the difference  – *F*0(*y*). Such an *EDF test statistic* typically has the form

.

Here *ψ*(*y*) is a weighting function. Special cases are the *Cramér-von Mises test statistic*:



where *ψ*(*y*) = 1, and the *Anderson-Darling test statistic*:



where *ψ*(*y*)= [*F*0(*y*)(1 *– F*0 (*x*))]-1.

At first sight such GOF test statistic seem difficult to calculate involving a nasty integration. In fact things are much more straightforward than this. For example, the Anderson-Darling statistic (which is conventionally denoted by *A*2) is easily calculated using the equivalent formula



where  is the value of the fitted CDF at the *i*th 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 4 which guarantees that  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 8.1 illustrates the process involved in calculating a GOF test statistic for the parametric case. Two cases are shown. In both cases the distribution from which **Y** has been drawn is assumed to be , but where  is unknown; thus in each case  has been fitted to the random sample **Y** giving the ML estimate  of . However in the first case  is the correct model. Thus  will be the EDF of a sample drawn from  which will therefore converge to this distribution. In the second case the true model, , is different from , and may even involve a set of parameters  that is different from . Thus the EDF  in this second case will *not* converge to , but to . 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  and  is *smaller* than the difference between  and the unknown true. This is because the fitted distribution  will follow the sample more closely than  because it has been fitted to the sample. This has to be allowed for in calculating the null distribution of the test statistic.

Figure 8.1: Process Underlying the Calculation of a GOF Test, *T*







*T*



Null Case: Fitted model  is the correct model.







*T*



Alternative Case: Fitted model, , is an incorrect model.

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.

**9 Bootstrapping a GOF statistic**

If we could obtain many values of *T* using the null case calculation of the GOF statistic depicted in Figure 8.1, then the EDF of the *Ti* 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  by its MLE . This is precisely the parametric bootstrap process. We provide some details of the parametric bootstrap in the next subsection but these will not be discussed in the lectures. More details can be found in Cheng and Currie (2009).

All being well  will be close in value to . Thus the distributional properties of the  will be close to those of a set of obtained under the null case calculation of Figure 8.1.

The parametric bootstrap method as it applies to a GOF statistic is illustrated in more detail in Figure 9.1, where  is the EDF of the bootstrap sample , and  is the bootstrap MLE of obtained from the bootstrap sample 

**Example 9.1:** Examine 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.

[Gamma Fit Toll Booth Data](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\GammaFitTollboothData.xls) [Normal Fit Toll Booth Data](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\NormalFitTollBoothData.xls) □

Figure 9.1: Bootstrap Process to Calculation the Distribution of a GOF Test, *T*











*i* = 1,2, ..., *B*

**9.1 (Extension) The Parametric Bootstrap**

There are two versions of bootstrapping – the simple bootstrap and the parametric bootstrap. As the parametric bootstrap is of particular relevance to GOF tests, we only describe parametric bootstrapping here.

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 9.2.

Figure 9.2: Parametric Bootstrap Process



**Y**2\*

*T*2\*

**Y**1\*

**Y***B*\*

.

.

.

.

*TB*\*

*T*1\*

□

**Part III Simulation Optimization**

# 10. Introduction

One of the primary goals of many simulation models is to find the system settings that result in optimal performance. For example

* Deciding between several different set-ups for a factory or hospital to maximize throughput
* Choosing the intervention or mix of interventions that leads to the least deaths when treating an infectious disease
* Optimizing the number of call centre staff on duty to minimize costs subject to constraints on response times

Using mathematical notation, we wish to minimize an output *f*(***x***), where ***x*** is a vector of decision variables and *f*(***x***) is the expected value of the random output *Y*(***x***),

*f*(***x***) = E***x***[*Y*(***x***)]. (10.1)

Nelson and Hong (2009) provide a useful classification for simulation optimization problems, dividing them into three main groups:

1. The feasible region for ***x*** has a small number of solutions; sufficiently small that it is possible to test all of them in the simulation model and select the solution with the best result.
2. The vector of decision variables, ***x***, is continuous.
3. The vector of decision variables, ***x***, is discrete and integer ordered.

The three examples given above come from each of these classes and are ordered in the same way.

In this course, we introduce ranking and selection algorithms that can be used to solve simulation optimization problems of type 1; stochastic approximation algorithms for the solution of problems of type 2 and random search algorithms for the solution of problems of type 3. For more detailed descriptions of all of these methods we would recommend the Handbook of Simulation Optimization edited by Michael Fu.

**11. Ranking and Selection Algorithms**

Ranking and selection algorithms are used when the number of options available is small, i.e. ***x*** can take only a few values, and it is possible to sample at each of these values. The difficulty lies in the fact that the output of the simulation model is stochastic and that we have only a limited time or computational budget for experiments.

We assume that each of the design points ***x***1, ***x*** 2, …, ***x*** *m* has a true output associated with it 1,  2, …,  *m*. When we run the simulation model, we make *ni* observations at each of the ***x*** *i* and obtain a set of sample averages E***x***[*Y*(***x****1*)], E***x***[*Y*(***x****2*)], …, E***x***[*Y*(***x****m*)]. These are approximations to the 1,  2, …,  *m*.

One of the best-known classes of ranking and selection algorithms are *Indifference Zone* procedures. Indifference zone methods aim to determine the number of observations that need to be made at each design point *ni*, *i* = 1, …, *m*, to ensure that the design point with the smallest output E***x***[*Y*(***x****k*)] has a true output *k*, that is within  of the **true** optimal value 0, such that 0 = min{*i* *i* = 1, 2, …, *m*}, with probability greater than 1 – . The parameter  is the significance level, e.g. 0.05. The variable  is known as the indifference-zone parameter and it is usually set to be the smallest difference that would be practically significant. Writing this more formally, we set the *ni* such that

P{ E***x***[*Y*(***x****k*)] > E***x***[*Y*(***x****i*)] | (*k*) – (*i*) ≥≥ 

The actual procedures used to determine the *ni* can be relatively complex and we do not go into details here.

**12. Stochastic Approximation Algorithms**

Stochastic approximation algorithms are used when the user inputs ***x***, are continuous. They can be thought of as the stochastic equivalent of steepest-descent algorithms, in that they use the gradient of the output *f*(***x***) to dictate the direction of movement of the algorithm.

Starting with an initial point ***x***0, the algorithm develops a sequence {***x****n*}, which converges to a unique (local) optimum, i.e. as *n* tends to infinity, ***x****n* will tend to the value of the user input that minimizes *f*(***x***).

There are two common algorithms on which many of the recent methods in this area are based: Robbins-Munro and Kiefer-Wolfowitz. Both have similar iterations and we only include the Kiefer-Wolfowitz equation below,

*xn*+1 = *xn* – *an*(*Y*(*xn* + *cn*) – *Y*(*xn* – *cn*))/2*cn*,

where *Y*(*x*) is obtained by running the simulation model with input parameters *x* and consequently is a stochastic variable. The rate of convergence of the algorithm to the optimum is dependent on the choice of the sequences {*an*} and {*cn*}, with the optimal choice being somewhat problem-dependent.

**13. Random Search**

Random search methods are iterative algorithms that move around the feasible space for the decision variable x in a random fashion. They are useful when the decision variables x are not continuous variables and the number of feasible options is greater than around 20.

At each iteration of the algorithm, one or more points will be sampled from the neighbourhood of the current point, and if a sufficient improvement is observed, the algorithm will move to this new point in the next iteration. We introduce a particular example of a random search method termed the Stochastic Ruler method, using the modifications introduced by Alrefaei and Andradóttir (2001).

The principle of the Stochastic Ruler method is that in each iteration we compare the value generated by the simulation model with a random value generated from the range of possible values for the objective function. This differentiates it from other random search algorithms in which the comparison is made between realisations of the objective function at different values of the decision variables.

Let E[f(**x**)] be the expected value of the objective function evaluated at ***x,*** where we write it as an expectation as it is the average over *p* iterations. We assume that we know the range of E[f(**x**)] and that it takes a minimum value of *a* and a maximum value of *b*, i.e. the optimal value will lie somewhere on the ruler between *a* and *b*. We assume that ***x*** can only take values in the space defined by **** and we set up a starting point for the algorithm ***x***0 ∈****.

At each iteration of the algorithm, we choose a new point ***x***’ from the *neighbourhood* of the current point ***x****n*. The neighbourhood, *N* must be defined such that it is equally likely to select ***x***’ when at ***x****n* as to select ***x****n* when at ***x***’. Examples of neighbourhoods could be any of the other possible values of ***x*** or perhaps the values of ***x*** that are one different from the current value. Unlike for other random search methods, the choice of neighbourhood structure is not critical to the convergence properties of the algorithm.

**Algorithm**

1. Define *a*, *b*, ***x***0 and the neighbourhood structure *N*
2. Generate a candidate solution ***x***’ from the neighbourhood of the current value *N*(***x****n*, .)
3. For *i* = 1 to *Mn*

Run the simulation model *p* times to obtain E[f(**x**)]

Generate a random number *U* from the uniform distribution *U*(*a*, *b*)

If E[f(***x***’)] > *U* then

***x****n*+1 = ***x****n* (i.e. stick with the current solution)

Exit for

End if

Next *i*

***x****n*+1 = ***x***’ (i.e. we like this solution, move to this point)

1. *n* = *n* + 1, Go to 1

Results suggest that using 1 ≤ *M* ≤ 5 and *p* = 1 results in a reasonable performance. The exact choice of *M* will depend on the neighbourhood structure chosen. If the neighbourhood *N* is chosen such that few local solutions are generated, a large *M* is recommended, and vice versa.

**Experimental Design and Analysis**

**Part IV Experimental Design and Analysis of Linear Models**

One of the most useful statistical models is the ‘Linear Model’. We encountered it in Example 2.2 of Part I. Here in Part IV of Experimental Design and Analysis, we discuss its form in more detail and examine how to fit it to data, including simulation data. We then give an introduction to how simulation experiments using this metamodel should be designed. In complex simulations, a well designed experiment is not just a luxury, but can mean the difference between a project that is successfully carried out in a *timely* way, and one that that is impossible to carry out in a realistic time-frame.

**14 Linear Regression Metamodels.**

Each observation is assumed to take the form (*Y*, *X*1, *X*2, … *XP*) where *Y* the output of interest, is assumed to be dependent on *P* *decision* or *explanatory*, variables *Xj*, *j* = 1, 2, …, *P*, often called *factors*. [Note that in this section we use the notation *P* for the number of factors, rather than the *M* previously used,] Assume there are *n* observations. In a simulation experiment this will have been obtained from *n* runs. In the linear (regression) model it is assumed that:

. (14.1)

where *Yi* is the *i*th observation and *Xij* is the observed value of the *j*th factor in the *i*th observation. It is usual to assume that the errors are mutually independent and normally distributed:

 ~ . (14.2)

The coefficients , *j* = 0, 1, …, *P* are assumed unknown and have to be estimated from the data. This is usually termed ‘fitting’ the model. We shall discuss this process in the next section. Notice that these coefficients appear linearly in (14.1), which is why the model is termed ‘linear’. At first sight it might appear that the factors also enter linearly, but this is not necessarily so. In the simplest case, where the factors are all distinct and quantitative (as opposed to being qualitative), the factors can indeed be regarded as appearing linearly, as each variable is present linearly and in its own right. In the general literature on the statistical design of experiments such a model is called a *main-effects model*.

However there is no difficulty in including quadratic, or more general second-order effects factor effects such as *X*12or *X*1*X*2. Thus the *second-order model*

 (14.3)

is *still* a linear model. Second order effects like *X*1*X*2 are often called *two-way interactions*. Indeed much of the statistical literature on the linear model develops the theory in terms of two-way interactions, rather than in terms of second-order regression models, to the extent that the first time reader may not realise that the two forms of model are identical. A curious, and rather unsatisfactory, aspect of the interaction approach is that often the pure quadratic terms *Xi*2 are omitted from consideration. It would seem inadvisable not to at least consider the possible effect of pure quadratic terms if two-way interactions are expected, or turn out to be important.

In complex simulations a simple initial approach is to assume that a linear model can be fitted to the results of a set of simulation runs, where we are free to select the factor values to be used in each of the simulation runs. Badly chosen factor values can result in difficulties in fitting the assumed linear model. Thus we may end up with a situation where the coefficients  are not accurately estimated or worse still where some cannot be estimated at all from the simulation runs made. To make this more precise, we first summarize how the coefficients can be estimated.

When there are a large number of explanatory variables it is useful to use vector notation. We can write (14.1) either in the partial vector form

*Yi*  = **X***i***b**+ *ei,* *i =* 1, 2, ..., *n* (14.4)

where **X***i* = (1, *Xi*1, *Xi*2, ,*XiP*) is a row vector, and the unknown coefficients are written as the column vector

**b***=* (*b*0*, b*1*,* ...,  *bP*)T (14.5)

(where the superscript T denotes the transpose), or in the full vector form

 (14.6)

i.e.

**Y** = **Xb** + **e** (14.7)

The values of the explanatory variables written in the matrix form, **X**, is called the *design matrix.* In classical regression this is usually taken to be non random. This is in line with our simulation viewpoint that the explanatory variables are decision variables who values we can choose and fix.

**Exercise 14.1:** Identify the observations **Y** and the design matrix **X**for the Fire Rescue Service Simulation Data. [ANOVAFireRescueServiceEG.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVAFireRescueServiceEG.xls)

**15 Fitting and Assessing the Linear Model**

**15.1 Least Squares Estimation**

The coefficients **b** are assumed unknown. They have to be estimated from the observations. The most commonly suggested method of estimation is that of *least squares* (LS). A good description of least squares estimation at a reasonably accessible level is Wetherill (1981). A good Web reference is <http://www.statsoftinc.com/textbook/stmulreg.html>. We summarize the method here.

Least squares (LS) estimation obtains **b** by minimizing the *sum of squares*

 (15.1)

with respect to **b**.An arguably more statistically satisfactory method is that of *maximum likelihood*, introduced in Part II. This latter method requires an explicit form to be assumed for the distribution of , such as the normal distribution of (14.2), whereas least squares does not require this assumption. In the special case where the errors are assumed to be normally distributed then least squares and maximum likelihood methods are in fact essentially equivalent.

The LS estimator of **b** is

 = (**X**T**X**)−1**X**T**Y**.(15.2)

The least squares method does not yield an estimate of , the variance of , and usually a quite different idea, that of *unbiasedness*, is invoked to provide an estimate. In contrast the maximum likelihood method provides an estimate directly as

 (15.3)

where

. (15.4)

Geometric Interpretation of fitting 



**y**

**1**

**x**

x1, y1

x3, y3

x2, y2

a.**1**

b **x**

**e**

**y** = a.**1** + b.**x** + **e**

**15.2 ANOVA**

The accuracy and effectiveness of the linear model in explaining the dependence of observed output *Y* on the explanatory variables can be neatly described by carrying out an *analysis of variance*. This is done as follows.

The *total sum of squares (corrected for the overall mean)* is

 (15.5)

where . This decomposes into

 (15.6)

where

 (15.7)

is called the *regression sum of squares*. This measures the reduction in total sum of squares due to fitting the terms involving the explanatory variables. The other term

 (15.8)

is called the (minimized) *residual or error sum of squares* and gives the variance of the total sum of squares not explained by the explanatory variables.

The sample correlation, , between the observations *Yi*  and the estimates  can be calculated from the usual sample correlation formula and is called the *multiple correlation coefficient*. Its square turns out to be

. (15.9)

This is called the *coefficient of determination*. *R2* is a measure of the proportion of the variance of the *Y*'s accounted for by the explanatory variables.

The sums of squares each have an associated number of *degrees of freedom* and a corresponding *mean square*

(i) For *SST:* *dfT* = *n* − 1, and *MST = SST/dfT*

(ii) for SSR: *dfR* = *P* (i.e. the # of coefficients − 1) and *MSR = SSR/dfR*

(iii) For SSE: *dfE* = *n* − *P* − 1 and *MSE =SSE/dfE*

Thus we have

*dfT = dfR + dfE*. (15.10)

Under the assumption that the errors, *ei*, are all independent and normally distributed, the distributional properties of all the quantities just discussed are well known.

If in fact the *bi  i =* 1,2,..., *P*  are zero so that the explanatory variables are ineffective then the quantity

 (15.11)

has the *F-distribution with P and* (*n* − *P* −1) degrees of freedom. If the *bi* are non zero the *F* tends to be larger.

These calculations are conventionally set out in the following analysis of variance (ANOVA) table

—————————————————————————————————

Source Sum of df MS F P

Squares

—————————————————————————————————

Regression *SSR P SSR/P MSR/MSE p-value*

Error *SSE* (*n* − *P* −1) *SSE*/(*n* − *P* −1)

—————————————————————————————————

Total *SST* (*n* −1)

—————————————————————————————————

**Exercise 15.1:** Carry out the ANOVA for Fire Rescue Service Simulation Data [ANOVAFireRescueServiceEG.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVAFireRescueServiceEG.xls)

**15.3 Individual Coefficients**

Either the coefficient of determination, *R* 2, or the *F - ratio* gives an overall measure of the significance of the explanatory variables. If overall significance is established then it is natural to try to identify which of the explanatory variable is having the most effect. Individual coefficients can be tested in the presence of all the other explanatory variables relatively easily.

Again we assume that the errors *ei* are *N*(0, *σ*2) variables. Then the covariance matrix of  is given by

Var() = (**X**T**X**)−1*σ*2. (15.12)

An alternative estimate of *σ* 2 is given by the *MSE*:

. (15.13)

If the true value of *bi* = 0, so that the explanatory variable is not effective, then it is known that

 (15.14)

where is the standard error of  has the *t-distribution with* (*n − P −* 1) *degrees of freedom*.

The P-value of *tj* i.e. Pr(*T* *>* |*tj*|) where *T* is a random variable having the t-distribution with(*n − p −* 1) degrees of freedom can then be found.

Alternatively, and rather better, is to calculate a 100(1 *− α*)%confidence interval for the unknown true value of *bj* as

 ± *t*(*α/2*). . (15.15)

**Exercise 15.2:** Identify  and  for the Fire Rescue Service Simulation Data.

[ANOVAFireRescueServiceEG.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVAFireRescueServiceEG.xls)

**15.4 Selecting Important Coefficients**

A simple method for identifying good possible models is to use bootstrapping. We generate *B* bootstrap samples of form (14.7). The simplest way to do this is to use parametric bootstrapping. Thus the *i*th bootstrap sample is

(15.16)

where each entry in the vector of errors , is a normally distributed variate sampled form the fitted error distribution .

For each bootstrap sample we estimate the coefficients using (15.2). Then for that sample we take as our best fitted model for that sample, that which includes just those coefficients *bj* for which the corresponding *tj* value is significantly different from zero.

It turns out , for an orthogonal design, that if the level of significance for testing each individual coefficient is taken as 91.7%, then overall, the method becomes equivalent to the criterion ‘choose that model for which the Mallows statistic *Cp* (introduced in Part III) is minimized’. We use this criterion even if the design is not orthogonal, as we are only interested in using bootstrapping to generate a set of ‘promising’ models that *might* be a good fit.

A method for choosing between these promising models is to calculate, using the *original* sample, the *Cp* statistic proposed by Mallows (1973) for each promising model. For a given model, *m* say, containing *p*  1 of the full set of *P* factors (thus when we include the unknown constant, there are *p* unknown coefficients), *Cp*  is defined as



where  is the variance estimated by fitting the full model with *P* factors, and  is the variance estimated from the model *m*. An alternative statistic is the Akaike Information Criterion (Akaike, 1970), which for the linear model reduces to

.

Asymptotically *Cp* and *AIC* have the same distribution. However *Cp*is perhaps more satisfactory for our purpose because of its ease of interpretation. Mallows (1973) shows that if the model *m* (with *p*  1 factors) is satisfactory in the sense that it has no bias, then the expected value of *Cp* is close to *p*:

.

Thus once all important factors are included, *Cp* will increase linearly with *p*. However if not all important factors are included, the expected value of *Cp* will be larger than *p*. A simple selection method is therefore the following.

**Bootstrap “Min *Cp*” Selection Method**

(i) Consider each of the models of selected by the simple bootstrap analysisand for each model m calculate Cp(m).

(ii) Select as the best model that *m* for which *Cp*(*m*) is minimum, with the expectation that this model will be satisfactory if .

The advantageof this bootstrap method of selecting a model is that it avoids having to examine a potentially huge number of possible models, but considers a set of good models produced by the bootstrap process.

For more details of the method see Cheng (2009).

**Exercise 15.2:** Suggest a model for the Fire Rescue Service Simulation Data. [ANOVAFireRescueServiceEG.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVAFireRescueServiceEG.xls)

**16 Prediction with the Linear Model**

The linear model is a particularly simple form of *regression model* where the regression function, E(*Y*(**X**)), is linear in the parameters. This allows a simple application of the analysis of Section 15 to predict the expected value of *Y*, E(*Y*(**X**0)), at any given value of the explanatory variables, **X**0 = (1, *X*01, *X*02, ,*X*0*p*), using

. (16.1)

An estimate of the accuracy of this prediction is given by

, (16.2)

where  can be estimated by either  of (15.3) or by  of (15.13).

Time is not explicitly involved in these formulas. Thus we are *not* using the word *predict* necessarily in the sense of forecasting the future. We use 'predict' only in the sense that the fitted regression model is used to *estimate* the value of regression function that corresponds to a particular set of values of the explanatory variables. If however one of the explanatory variable is time, *t*, then we *can* use linear regression as a forecasting tool.

**17 Additional Explanatory Variables**

In any regression model, one may consider introducing additional explanatory variables to explain more of the variability of *Y*. This is especially desirable when the error sum of squares, *SSE* is large compared with *SSR* after fitting the initial set of explanatory variables.

One useful type of additional variable to consider is what are called *indicator* variables. This arises when one wishes to include an explanatory variable that is *categorical* in form. A categorical variable is one that takes only a small set of distinct values.

For example suppose we have a categorical variable, *W*, taking just one of four values Low, Medium, High, Very High. If the effect of *W* on *Y* is predictable then it might be quite appropriate to assign the values 1, 2, 3, 4 to the categories Low, Medium, High, Very High and then account for the effect of *W* using just one coefficient *a*:

*Yi*  = *b*0 *+ b*1*Xi*1 *+ b*2*Xi*2 +...+ *bp Xip* + *aW* + *ei,* *i =* 1, 2, ..., *n*. (17.1)

However if the effect of each of the different possible values of the categorical variable on *Y* is not known then we can adopt the following different approach. If there are *c* categories then we introduce (*c* − 1) indicator variables. In the example we therefore use (4 − 1) =3 indicator variables, *W*1, *W*2, *W*3. The observations are assumed to have the form

*Yi*  = *b*0 *+ b*1*Xi*1 *+ b*2*Xi*2 +...+ *bp Xip* + *a*1*Wi*1+ *a*2*Wi*2+ *a*3*Wi*3 + *ei,*

*i =* 1, 2, ..., *n* (17.2)

where

 (17.3)

Note that for each point *i* only one of *Wi*1, *Wi*2, *Wi*3 is equal to unity, the other two being zero.

Note also that an indicator variable is *not* needed for the final category as its effect is absorbed by the overall constant *b*0.

**18 Experimental Designs**

The topic of experimental design concerns the important issue of selecting the values of the explanatory variables to be used in making each run of an overall simulation experiment. There are many useful designs, depending on aim of the simulation experiment. We shall only consider a few of the most useful designs. Useful brief introductions are given by Kleijnen (2007) and Sanchez (2008).

There is one useful point to note when considering designs. Equation (15.12) giving the variance-covariances of the estimates of the coefficients in terms of the design matrix **X**,is a very useful formula. Firstly it provides a simple test for checking that all the coefficients which are to be included in the model can actually be estimated using the proposed design. All that is required is to check if the matrix **X**T**X** appearing in (15.12) is invertible, this condition being both necessary and sufficient to ensure that all the coefficients in the vector **b** can be estimated. If all the coefficients can be estimated, the diagonal entries in the inverted matrix, (**X**T**X**)-1, being simply the variances of the estimators of the coefficients multiplied by , can then be used to check how well individual coefficents can be estimated. This allows different competing designs to be easily compared.

**18.1 Main Effects Model**

**18.1.1 Factorial Designs**

Consider the main effects model (14.1). Here the value of any given *bj* could be estimated by making two runs with *Xj* set to different values in the two runs, whilst the other factors are unchanged in the two runs. This is the so-called method of varying values one at a time. This method is not usually very efficient compared with what are called (full) *factorial designs*. The simplest of these is where it is assumed that each factor can take just one of two possible values. These are conventionally assumed to be standardized so as to be +1 and 1, representing ‘High’ and ‘Low’ settings. Designs are represented in the literature using such normalized settings. In a full factorial experiment all possible combinations of factors settings at these two levels are taken, yielding 2*p* design points in all. Table 18.1 shows the 8 design points of a three factor experiment.

Table 18.1: Variable settings for 23 factorial design

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Variables

Design point X1 X2 X3

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 1 1 1

2 +1 1 1

3 1 +1 1

4 +1 +1 1

5 1 1 +1

6 +1 1 +1

7 1 +1 +1

8 +1 +1 +1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

We give two examples. The first involves just two factors.

**Example 18.1.** Figure 18.1 depicts the runs made in a one-at-a-time experiment involving two factors, and the corresponding full 22 factorial experiment. [ANOVATwoMainEffects.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVATwoMainEffects.xls) compares the effectiveness of these two designs.

X1

X2

X2

X1

Figure 18.1 One at a time Experiment and Full Factorial Experiment for 2 Factors

([ANOVATwoMainEffects.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVATwoMainEffects.xls)) It will be seen that in the one-at-a-time experiment the estimate of *b*0 has variance  whilst the estimates of the other coefficients all have variance . In contrast, in the full factorial experiment the estimates of all the coefficients have a smaller variance, . Moreover the design is *orthogonal* so that the coefficient estimates are all independent. This is a very useful property for a design to have as it means that contribution of a particular explanatory variable does not depend on those of other explanatory variables. Full factorial designs are always orthogonal.

A full factorial experiment, when each factor is considered at two levels, has 2*p* design points. This is often provides far more design points than we need to estimate all the unknown coefficients, including . A neat solution is to use a *fractional factorial experiment*, where only a given proportion of the full factorial design is used. An introduction to factorial designs is given in Law (2007). A typical fractional factorial design based on a 2*p* full factorial design contains 2*p – m* design points, with *m* chosen to ensure that main effects can be estimated adequately using just the fractional factorial design with its reduced number of design points. (The notation 2*p – m* is standard to indicate the number of factors and the fraction of the full design being used.)

Half fractions are particularly easy to generate. To create a 2*p* – 1 factorial design, first set up the *p* columns ( this includes the *X*0 column) of a full factorial experiment for (*p* 1) factors. Then add one last column, (corresponding to the final *p*th factor) by multiplying together all the entries for the first (*p* 1) factors.

**Example 18.2.** Figure 18.2 shows the designs for a one-at-a-time experiment and a *fractional factorial experiment* using a half-fraction, when there are three factors each being considered at just two levels. Notice that the basic one-at-a-time experiment has (*p* + 1) design points. This enables the coefficients to be estimated but not the variance . We compare this with the fractional factorial 23-1, as shown in the Figure 18.2.

X1

X3

X2

X1

X3

X2

Figure 18.2 One at a time Experiment and 23-1 Fractional

Factorial Experiment for 3 Factors

[ANOVAThreeMainEffects.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVAThreeMainEffects.xls) compares the two experiments in this example. Note that as both designs has only 4 design points each, we cannot estimate  as well as all the coefficients **b**, using either design. In the spreadsheet we simply replicate each design in its entirety to provide extra observations with which to estimate .

**Exercise 18.1:** Make this comparison using the spreadsheet.

**18.1.2 Plackett-Burman Designs**

Even with the use of fractional factorial designs, it is often the case that a fractional factorial design cannot be found with a number of design points closely matched to the number of factors. Where a precise match *is* obtained the design is said to be *saturated*. Where main effects are concerned, Plackett-Burman (Plackett and Burman, 1946) designs are generalisations of fractional factorial designs which exist for *n* = 4, 8,…, There is no single comprehensive method for finding a design for *n* = 4*t* for all *t*, but the list is complete up to *n* = 100, and nearly so up to 200. Plackett-Burman designs are orthogonal.

**19 Interactions**

Fractional factorial and Plackett Burman designs enable main effects to be efficiently estimated assuming that interactions are negligible. However there may be uncertainty about this. A particularly serious concern is the problem of *confounding*. This is the situation where an interaction cannot be separated from a given main effect or from a given lower order interaction that is known to be important. Two such quantities which cannot be separately estimated in a given design are said to be *confounded* or *aliased*. Designs are usually classified according to the form of aliasing that they are subject to. A particular useful form of classification is the *resolution* of a design.

A design of resolution *R* (often denoted by Roman numerals) is one where an *l*-way interaction (with a main effect defined as a *1-way interaction*) may possilby be confounded with an interaction of order *R-l*, but not one lower. Thus in a resolution III design, a main effect (i.e. where *l*  = 1) may be confounded with an interaction of order 3*-*1 = 2 (i.e a two-way interaction) but not with another main effect. A resolution III design will therefore estimate main effects properly but only if it can be assumed that all two-way (and higher order) interactions are negligible. For a resolution IV design, a main effect may be confounded with an interaction of order *R-*1 = 4-1 = 3, but not with any two-way interactions. However a two-way interaction may be confounded with an interaction of order *R-l* = 4*-*2 = 2. Thus, the two-way interactions may be confounded with one another.

**Exercise 19.1.** What resolution is required to be able to estimate all the coefficients of a quadratic regression model?

**20 Central Composite Designs**

Designs with factors at only two levels cannot be used to assess all second order effects. 3*p* factorial experiments can be used, but can lead to an unnecessary increase in the number of design points being added to an experiment.

A simple alternative is to use what are called *central composite designs.* Start with a 2*p* or a Resolution 5 2*p-m* fractional factorial, then add a centre point and two ‘star Figure 20.1 Construction of a Central Composite Design

X1

X3

X2

Fractional Factorial

+ Star Additions =

Central Composite

points’ for each factor. The centre point is simply *X*0 = 1, *X*1 = 0, *X*2 = 0, … *Xp* = 0. The star points are *X*0 = 1, *X*1 = 0, *X*2 = 0, …, *Xi* = *c*, … , *Xp* = 0 and *X*0 = 1, *X*1 = 0, *X*2 = 0, …, *Xi* = +*c*, … , *Xp* = 0, for *i* = 1, 2, …, *p*, where we can take *c* = 1, though other values are possible. Figure 20.1 depicts the case for three factors.

**Example 20.1**. [ANOVAWSC04Optim.xls](file:///C:\Users\ccurrie\Documents\Russell%20Research\NATCOR\Examples\ANOVAWSC04Optim.xls) contains an example discussed in Cheng and Currie (2004) involving a simulation model of car performance depending on six factors. The objective was to find settings of the six factors to optimize the tuning of the car. The experiment therefore involves fitting a quadratic regression model to data obtained from simulation runs and then using the regression model to estimate the optimum (minimum) point.

**21 Comments on Design of Experiments**

The subject of design of experiments is a large one and we have only been able to touch on some of the main features. For further introductory material see Kleijnen (2007) and Sanchez (2008).

Though there is a large theory, the special circumstances of a particular practical application may still make it difficult to choose a good design. The spreadsheets provided in this course do however show how any given design can be rapidly assessed for how effective it is for estimating factors of interest, enabling designs to be set up and modified interactively.

**22 Final Comments**

This course 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.

Examples have been drawn mainly from regression and ANOVA applications. These have been for illustration only and I have not attempted to survey the range of statistical models likely to be encountered in practice, where typically different aspects of modelling 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 logisitc regression model but with a linear predictor involving the categorical factors. Though this example has not been discussed explicitly in this course, all the elements needed to analyse 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 course.

In fact the spreadsheets given for the examples in this course 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.

**References**

Alrefaei, M.H. and Andradóttir, S. A Modification of the Stochastic Ruler Method for Discrete Stochastic Optimization. *European Journal of Operational Research*. 133, pp 160-182.

Banks, J. (1998). Handbook of Simulation. New York: Wiley.

Cheng, R.C.H. and Currie, C.S.M. (2004) Optimization by Simulation Metamodelling Methods. In *Proceedings of the 2004 Winter Simulation Conference, eds R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters*. Piscataway: IEEE, 486-490

Cheng, R.C.H. and Currie, C.S.M. (2009) Resampling Methods of Analysis in Simulation Studies. In *Proceedings of the 2009 Winter Simulation Conference, eds M.S. Rossetti, R.R. Hill, B. Johansson, A. Dunkin and R.G. Ingalls.* Piscataway: IEEE, 45-59.

Cheng, R.C.H. (2009, In press) Computer Intensive Statistical Model Building. In *Advancing the Frontiers of Simulation: A Festschrift in Honor of George Samuel Fishman*  Springer, pp 47 -66.

Chernick, M.R. (1999). Bootstrap Methods, A Practitioner's Guide. New York: Wiley.

D'Agostino, R.B. and Stephens, M.A. (1986). Goodness of Fit Techniques. New York: Marcel Dekker.

Davison, A.C. and Hinkley, D.V. (1997). Bootstrap Methods and their Application. Cambridge: Cambridge University Press.

Fu, M.C. ed. (2015). Handbook of Simulation Optimization. Springer-Verlag, New York.

Kleijnen, J.P.C. (2007). Regression Models and Experimental Designs: A Tutorial for Simulation Analysis. In Proceedings of the 2007 Winter Simulation Conference, eds S.G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J.D. Tew and R.R. Barton. Piscataway: IEEE, 183-194.

Krzanowski, W.J. (1998). An Introduction to Statistical Modelling. London: Arnold.

Law, A.M. and Kelton, W.D. (1991). Simulation Modeling and Analysis, 2nd Ed., New York: McGraw-Hill.

Law, A.M. (2007). Simulation Modeling and Analysis, 4th Ed., New York: McGraw-Hill.

Nelson, B.L. and Hong, L.J. (2009). A Brief Introduction to Optimization via Simulation. In *Proceedings of the 2009 Winter Simulation Conference, eds M.S. Rossetti, R.R. Hill, B. Johansson, A. Dunkin and R.G. Ingalls.* Piscataway: IEEE,75-85.

Plackett, R.L. and Burman, J.P. (1946). The Design of Optimum Multifactor Experiments, Biometrika, 33, 305-325.

Sanchez, S.M. (2008). Better than a Petaflop: The Power of Efficient Experimental Designs. In Proceedings of the 2008 Winter Simulation Conference, eds S.J. Mason, R.R. Hill, L. M­önch, O. Rose, T. Jefferson and J.W. Fowler. Piscataway: IEEE, 73-84.

Urban Hjorth, J.S. (1994). Computer Intensive Statistical Methods. London: Chapman & Hall.

Wetherill, G.B. (1981). Intermediate Statistical Methods. London: Chapman and Hall.

**Worksheet 1**

**Examining the GammaFitTollboothData.xls and RegressionFitMorocco.xls Excel Spreadsheets**

1. Open the GammaFitTollboothData.xls Excel worksheet file.
2. Go to the Data sheet and examine the format of the data.
3. Do the following to examine some of the Visual Basic Programs attached to this file.
   * 1. If the Visual Basic Tool bar is not activated then go to View>Toolbars>Visual Basic
     2. Click on the Visual Basic Editor icon on the VB Tool bar. This opens the Visual Basic Macros/programs page.
     3. If the 'Project Explorer' window is not already displayed on the screen then open it by going to View>Project Explorer. This window shows a list of the worksheets (Excel Objects) and a list of the Modules. Click on the + sign next to the word Modules to open the Modules file and display a list of all the names of the individual modules attached to this excel file. Double clicking on the name of an individual module in this list brings up the programs belonging to that module.
4. Double click on the ‘LogLikelihoodCalc’ module. This displays all the programs /functions involved in calculating the loglikelihood.
5. Scroll down to the LogPdf() function. This contains the log pdf of the distribution fitted to the data. In this example it is a gamma pdf with two parameters: Alpha = Par(1) and Beta = Par(2). [What is the mean and variance of the distribution?]
6. The spreadsheet also calculates a performance curve (rather like, but not actually a regression function, why not?) this is done in the module ‘PerfIndexandCI’. Double click on the name of this module in the list of modules and this brings up the routines in the module. The Eta() Function at the end of this module is the Performance Index. Examine the form of this function and verify that it is the one mentioned in the Lecture Notes. This function is calculated and confidence intervals calculated for it using the control buttons on the Performance Index worksheet.
7. Examine the RegressionFitMorocco.xls spreadsheet in the same way.
8. The ‘LogLikelihoodCalc’ module contains the logpdf function used in the regression model. Check that you can see what the difference is compared with the toll booth example.
9. You should have noted in the previous step that the regression function Eta() is used explicitly in the calculation of the logpdf. The regession function Eta() is immediately below the logpdf function. Verify that it is the one mentioned in the lecture notes.

**Worksheet 2**

**FITTING A SUITABLE MODEL TO THE ‘TrafficQueueEG’ DATA**

Open the RegressionFitMorocco.xls excel worksheet file.

1. Go to the Data sheet and delete any data that is in columns A and B.
2. Open the TrafficQueueEG.xls excel worksheet file.
3. Copy the data in Sheet1, cells A3-B33 of the TrafficQueueEG file.
4. Paste this traffic data into the Data sheet of the RegressionFitMorocco file, so that the headings go in cells A10 & B10, and the data starts in cells A11 & B11.
5. Type the number of observations in this traffic data into cell 9B.
6. Save this amended version of the RegressionFitMorocco.xls file as RegressionFitTraffic.xls. (File>Save As…)

We will now try fitting a suitable model to this traffic data. To do this we need to edit some of the Visual Basic Programs attached to this file.

* + 1. If the Visual Basic Tool bar is not activated then go to View>Toolbars>Visual Basic
    2. Click on the Visual Basic Editor icon on the VB Tool bar. This opens the Visual Basic Macros/programs page.
    3. If the 'Project Explorer' window is not already displayed on the screen then open it by going to View>Project Explorer. This window shows a list of the worksheets (Excel Objects) and a list of the Modules. Click on the + sign next to the word Modules to open the Modules file and display a list of all the names of the individual modules attached to this excel file. Double clicking on the name of an individual module in this list brings up the programs belonging to that module.
       1. Double click on the ‘LogLikelihoodCalc’ module. This displays all the programs /functions involved in calculating the loglikelihood.
       2. Scroll down to the LogPdf() function. This contains the model function that we will fit to our data. It is a Normal pdf with mean = Eta() and standard deviation = Par(1).
       3. Now look at the Eta() Function below. This contains the function that we will assign to the model mean.

We must consider what type of function is appropriate for Eta() and whether it is valid to assume constant variance for the model. Take another look at the plot for the traffic data. Does the variance in the data seem to be constant? No? Then propose a suitable function for the variance. For example functions of the form:

θ1 x / (1-x)^θ2 , θ1 + θ2 x (linear function) , θ1 + θ2 x + θ32 x (quadratic function), etc...

* + - 1. Return to the RegressionFitTraffic excel file and the LogLikelihood module. At the moment the variance is programmed to be constant, Par(1). To create a variance that is dependent on the traffic intensity type in this Visual Basic function above the Function LogPdf(). ( NB. ‘ signifies a comment. These are comments for your information and are ignored by the program.)

Public Function Par1function(ByRef X As Double, ByRef Par() As Double) As Double

‘ This contains the function for sigma = Par(1)

‘Change this function to change the standard deviation (sigma)

(*Type your proposed function for the standard deviation (sigma) here, for example..)*

Par1function = Par(1) \* X / (1 - X)^2

End Function

* + - 1. Now go to the LogPdf() Function and amend it so that where it previously called Par(1) it now calls Par1function(X, Par) i.e.

LogPdf = -0.5 \* (Log(2 \* PI) + Log(Par1function(X, Par) \* Par1function(X, Par)) \_

+ ((Eta(X, Par) – Y) / (Par1function(X, Par)))^2 )

* + - 1. There is one more place in the program that we must edit to allow for a non-constant variance. Double click on the module ParaBootstrapParMLERoutine in the Project Explorer window. This brings up the macro ParaBootstrapParMLE(). At the top of the page, after the green comment ‘Parametric BootStrap of MLE Pars programme, type in the declaration

Dim Par1 As Double

Scroll down till you see the green comment

‘------- Generate the bootstrap sample of Y’s ----. Amend the three lines of program immediately below this comment so that it reads:

For I = 1 To NumObsns

Par1 = Par1function(XOrig(I), ParMLE())

YBoot(I) = Eta(XOrig(I), ParMLE()) + WorksheetFunction.NormInv(Rnd(), 0, Par1)

Next I

* + - 1. The program will now fit a variance term that is dependent on X (traffic intensity). You can try different functions for the variance by going to the Par1function() Function in the LoglikelihoodCalc module and changing the line: Par1function = *(…type the new function here…)*
      2. What form should Eta() take?

Return to the LogLikelihoodCalc module and scroll down to the Eta() function. Comment out (using ‘) the function/s you do not want Eta to take and type in the function you have chosen.

For example,

Public Function Eta(ByRef X As Double, ByRef Par() As Double) As Double

‘type in your function for Eta here

‘remember not to use Par(1) as this is the parameter reserved for use in the ‘standard deviation.

Eta = (Par(2) + (Par(3) \* X))/((1 - X)^Par(4))

End Function

* + - 1. Now bring back up the excel worksheets and go to the Optimize sheet. Type in the total number of parameters you have now used for Par1function() and Eta() into cell B9. Click on the ‘Clear Iteration Information’ button to clear the screen. Type in a suitable set of initial values (for the Neldermead optimisation algorithm) into Col B starting at cell B11. (NB. Finding suitable starting values can sometimes require trial and error). For the above suggested Eta function, try initial values of b1 = 0.1; b2 = 0.1; b3 = 0.1 and b4 = 1.
      2. Press the ‘1:Optimize’ button. Remember to copy the MLE estimate values into the initial value cells and re-optimize until the estimates seem stable. Press ‘2:Hessian’, ‘3:Cov&Conf Intvls’ button to create confidence intervals.
      3. Go to the Fit sheet and press the ‘Calc Fitted Curve Details’ button and ‘Make Chart’ buttons to see how well your model seems to fit to the data.
      4. Go to the Bootstrap sheet. Press the Bootstrap button to create bootstrap parameter estimates. Then press ‘MakeScatterPlot’ button. Does an assumption of normality seen valid? (i.e. are the points fairly randomly scattered? Or if there is correlation, is it linear rather than non-linear, e.g. quadratic?) You can alter the number of bootstraps performed by altering the number in cell B7.
      5. Go to the RegressionFunction sheet. Enter the range of values for X in cells D7 and D8 e.g. 0.95 and 0.2. Press the ‘CalcAsympRegnCIs’ button and then the ‘Calc Bootstrap Regn CIs’ buttons. Look at the results graphically by pressing buttons ‘Make Asymp Regn CI Chart’ and ‘Make Bootstrap Regn CI Chart’. You can slow down the drawing of the Bootstrap chart by typing N into cell L9.
      6. Are you satisfied with your model fit? If not go back into the Visual Basic programs and try another function for Eta and/or the variance.

**Worksheet 3**

**FITTING A SUITABLE MODEL TO THE ‘CortisolAssay’ DATA**

1. Open the RegressionFitTraffic.xls excel worksheet file that you previously created in Worksheet 1. Clear the data in columns A and B of the Data sheet.
2. Open the CortisolData.xls excel worksheet file.
3. Copy the data in Sheet1, cells A3-B67 of the CortisolData file.
4. Paste this traffic data into the Data sheet of the RegressionFitTraffic file, so that the headings go in cells A10 & B10, and the data starts in cells A11 & B11.
5. Type the number of observations in this traffic data into cell 9B.
6. Save this amended version of the RegressionFitTraffic.xls file as RegressionFitCortisol.xls. (File>Save As…)

We will now try fitting a suitable model to this Cortisol Assay data. To do this we need to edit some of the Visual Basic Programs attached to this file.

1. If the Visual Basic Tool bar is not activated then go to View>Toolbars>Visual Basic
2. Click on the Visual Basic Editor icon on the Visual Basic Tool bar. This opens the Visual Basic Macros/programs page.
3. If the 'Project Explorer' window is not already displayed on the screen then open it by going to View>Project Explorer. This window shows a list of the worksheets (Excel Objects) and a list of the Modules. Click on the + sign next to the word Modules to open the Modules file and display a list of all the names of the individual modules attached to this excel file. Double clicking on the name of an individual module in this list brings up the programs belonging to that module.
4. Double click on the ‘LogLikelihoodCalc’ module. This displays all the programs /functions involved in calculating the loglikelihood.
5. Scroll down to the Par1function() Function. This contains the function for standard deviation of the model.
6. Look at the LogPdf() Function below. This contains the model function that we will fit to our data. It is a Normal pdf with mean = Eta() and standard deviation = Par1function.
7. Now look at the Eta() Function below. This contains the function that we will assign to the model mean.

We must consider what type of function is appropriate for Eta() and whether it is valid to assume constant variance for the model. Take another look at the plot for the Cortisol data.

1. Return to the RegressionFitCortisol excel file and the LogLikelihood module. If you think it is reasonable to assume a constant variance then type the line **Par1function = Par(1)** in the Par1function() Function so that the code looks like:

Public Function Par1function(ByRef X As Double, ByRef Par() As Double) As Double ‘ This contains the function for sigma = Par(1)

‘Change this function to change the standard deviation (sigma)

Par1function = Par(1)

End Function

( NB. ‘ signifies a comment. These are comments for your information and are ignored by the program.)

1. What form should Eta() take?

Scroll down to the Eta() function. Comment out (using ‘) the function/s you do not want Eta to take and type in the function you have chosen.

For example,

Public Function Eta(ByRef X As Double, ByRef Par() As Double) As Double

‘type in your function for Eta here

‘remember not to use Par(1) as this is the parameter reserved for use in the ‘standard deviation.

Eta = Par(2) + Par(3) \* X + Par(4) \* X \* X

End Function

This is a quadratic function. You can try some other functions of your own.

1. Now bring back up the excel worksheets and go to the Optimize sheet. Type in the total number of parameters you have now used for Par1function() and Eta() into cell B9. Click on the ‘Clear Iteration Information’ button to clear the screen. Type in a suitable set of initial values (for the Neldermead optimisation algorithm) into Col B starting at cell B11. (NB. Finding suitable starting values can sometimes require trial and error). Try this set of starting values for the above quadratic Eta function:

b1 = 385; b2 = 860; b3 = -680; b4 = 23.

1. Press the ‘1:Optimize’ button. Remember to copy the MLE estimate values into the initial value cells and re-optimize until the estimates seem stable. Press ‘2:Hessian’, ‘3:Cov&Conf Intvls’ button to create confidence intervals.
2. Go to the Fit sheet and press the ‘Calc Fitted Curve Details’ button and ‘Make Chart’ buttons to see how well your model seems to fit to the data.
3. Go to the Bootstrap sheet. Press the Bootstrap button to create bootstrap parameter estimates. Then press ‘MakeScatterPlot’ button. Does an assumption of normality seen valid? (i.e. are the points fairly randomly scattered? Or if there is correlation, is it linear rather than non-linear, e.g. quadratic?) You can alter the number of bootstraps performed by altering the number in cell B7.
4. Go to the Regression Function sheet. Enter the range of values for X in cells D7 and D8 e.g. 2.5 and -3.5. Press the ‘CalcAsympRegnCIs’ button and then the ‘Calc Bootstrap Regn CIs’ buttons. Look at the results graphically by pressing buttons ‘Make Asymp Regn CI Chart’ and ‘Make Bootstrap Regn CI Chart’. You can slow down the drawing of the Bootstrap chart by typing N into cell L9.
5. Are you satisfied with your model fit? If not go back into the Visual Basic programs and try another function for Eta (and/or the variance).

**Worksheet 4**

**TASK: Carry out an ANOVA analysis of the tyre data in TyreData.xls**

**Question: Which of the main and interaction effects are significant?**

1. Open the TyreData.xls excel workbook. Have a look at the data in Sheet1.
2. Open the Cement Data analysis spreadsheet named CementBootstrap-ANOVA.xls

In order to carry out an ANOVA analysis of this tyre data you can use the cement data program (CementBootstrap-ANOVA.xls) as a template. All that is required is to arrange the tyre data into a suitable format and insert it into the correct place in the Data sheet in the CementBootstrap-ANOVA.xls spreadsheet. Then you can run the analysis using the ANOVA sheet and the Bootstrap sheet to answer the question at the top of this sheet.

The following instructions will take you through the analysis process step by step.

**Analysis 1 (Basic)**

1. Go to the Data sheet in CementBootstrap-ANOVA.xls. Replace the ‘Heat’ data with the ‘Wear’ data; the X1 data with the ‘Position’ data; and the X2 data with the ‘Car’ data. Delete the columns for X3 and X4 as these are not required. Change the ‘Car’ values from A, B, C into numbers, e.g. 1, 2, 3. Extend the values in columns A, B and D to match the new number of observations. Remember to put the new number of parameters into cell C12 and the new number of observations into cell C11.
2. Save this amended version of CementBootstrap-ANOVA.xls as TyreBootstrap-ANOVA.xls.
3. Go to the ANOVA sheet and carry out the analysis of the tyre wear data.
4. Go to the Bootstrap sheet and carry out the bootstrapping of the MLE parameters. (Try both of the available bootstrap methods by changing the value in cell B6).

**Analysis 2**

* **To analyse the interactions of the various levels** of each variable it is necessary to re-enter the data in a different format.
* We require a separate column for each level of the position and car variables and a column for each interaction between the position and car levels.
* Remember that you have to set one of the levels for each variable as the base level. Therefore if you set the 1st levels (Position1 and Car1) as the base levels you will only need to create data columns for levels: Positions 2, 3, 4, Car 2 and 3.
  1. In Sheet ‘Data’, copy and paste the Position data column so there are 3 columns of Position data next to each other.
  2. Copy and paste the Car data column so there are 2 columns of Car data next to each other.
  3. Re-name the columns to read Position2, Position3, Position4, Car2 and Car3.
  4. Amend the data in Position**2** column: change all numbers **except 2s** into 0s. Then change the 2s into 1s.
  5. Amend the data in Position**3** column: change all numbers **except 3s** into 0s. Then change the 3s into 1s.
  6. Amend the data in Position**4** column: change all numbers **except 4s** into 0s. Then change the 4s into 1s.
  7. Do the same to the data in the two Car columns.
  8. In order to add interactions to the model create further columns of data by multiplying together the various Position and Car level columns.

1. Name the next free column (J) Pos2\*Car2.
2. In the first cell of data (J18) type in the formula = *Click on 1st cell of data in column Position2(E18)* \* *Click on 1st cell of data in column Car2(H18) Return*
3. Click back onto the cell you just entered the formula into (J18). Drag the contents of the cell down into the 35 cells below*. (To drag: Position the mouse indicator over the bottom left corner of the cell until it changes to a black cross. Hold the left mouse button down and move the mouse indicator down till you reach cell J53. Release the mouse button.)*
4. Repeat steps i. to iii. for all the other interactions e.g. Pos2\*Car3, Pos3\*Car2 …
   1. Remember to put the correct number of parameters (variables) into cell C12.
   2. Go to the ANOVA sheet and carry out the analysis of the tyre wear data.
   3. Go to the Bootstrap sheet and carry out the bootstrapping of the MLE parameters. (Try both of the available bootstrap methods by changing the value in cell B6).