UNIVERSITY OF CALGARY

Modeling ACD Data To Improve Computer Simulation of Call Centers

by

Sherry L. Oh

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF BUSINESS ADMINISTRATION

FACULTY OF MANAGEMENT

CALGARY, ALBERTA

JULY, 1999

0-612-47978-1

Abstract

More and more often, call center managers are turning to computer simulations to help give them operational performance insights. To model service time for simulation and to evaluate queueing theory assumptions of exponentially distributed service times, analysts require an accurate distribution of call handle time. Unfortunately, automated call distributors (ACDs) generally report only average call handle time by time block – providing the analyst with incomplete, *temporally aggregated* data. An often-used method to model this data is to assume an exponential distribution with the calculated overall mean. The use of this method immediately raises two questions: is an exponential service time still an accurate reflection of call handle time (service time) and if not, how can the true distribution be estimated from the aggregated data readily available from ACDs? This thesis presents several alternate methods to arrive at a call handle time distribution and analyzes the efficacy of each.

Table of Contents

List of Tables

List of Figures and Illustrations

Chapter 1 Introduction

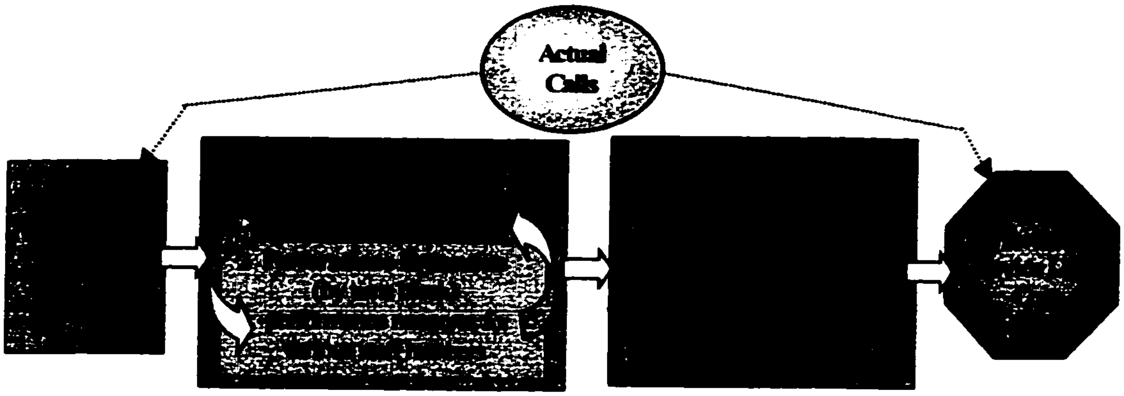
1.1 General Overview of Incoming Call Center Management

There are two general categories of call centers -- outgoing and incoming. This paper will focus on the operational challenges of the incoming call center and the input data analysis issues that analysts face in this environment.

Incoming call center management has been defined as "the art of having the right number of skilled people and supporting resources in place at the right times to handle an accurately forecasted workload, at service level and with quality" (Cleveland and Mayben 1997). This definition hints at the role that effective operations management/research techniques should play in the effective management of incoming call centers. With call centers employing millions of people in North America, the challenges of managing these centers are beginning to receive more academic attention.

A recent general overview of Call Center Operations can be found in an article in the Encyclopedia of Operations Research and Management Science, 2nd Edition (Grossman et al., 1999a). A more detailed description of call center operations is presented in a working paper (Grossman et al.,1999b), in which a five-stage approach to call center scheduling is proposed. These five stages are outlined in Figure 1 below. The 'time blocks' referred to in the figure represent blocks of time for which forecasts are prepared and schedules developed. There has been a trend in call centers towards using smaller (15 or 30-minute) time blocks to schedule TSRs. The acronym, TSR, has different meanings from business to business; it can stand for telephone/transaction/telecommunications sales/service/support representative (Anonymous 1996).

• Figure 1: Overview of Call Center Scheduling Process (Grossman et al. 1999b)



The scheduling process begins by obtaining a reliable forecast of the number of calls that will enter the call center for each time block in the scheduling horizon. From that point, staff requirements for each time block must be estimated. A variety of performance estimation techniques can be used for this purpose. Common methods include queueing theory applications such as 'Erlang tables' and 'Call Center calculators'.

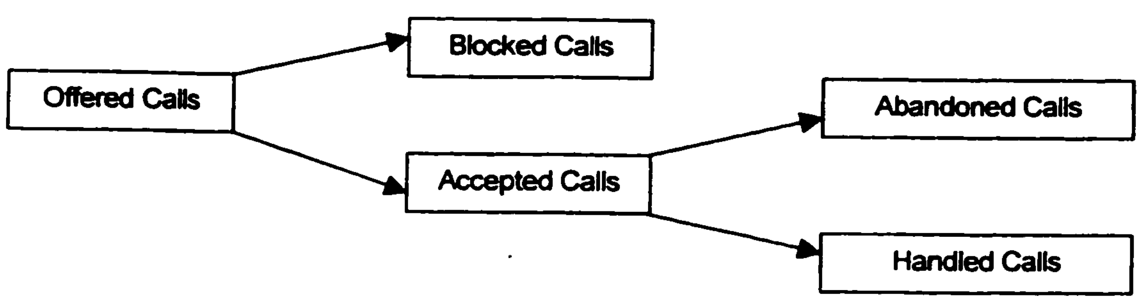
Recently, as call centers have adopted more complicated configurations such as skill-based routing, the limitations of queueing theory have led to a growing use of computer simulations to accurately estimate system performance. Call center performance is often described as a 'service level' -- the percentage of calls answered within a minimum acceptable time (e.g. 80% of calls answered within 30 seconds). The staff requirements stage is complete once it has been determined how many TSRs are required during each time block in order to achieve a target service level.

The next stage is to develop a schedule that will closely match the established staff requirements. Since each time block may have widely varying staff levels and call centers often must work within restrictive labour rules, it is often extremely difficult to achieve a 'perfect schedule' that will match the staff requirements determined in the previous stages. The final stage of rostering matches actual employees to the scheduled shifts (or 'tours' of shifts), producing a final employee schedule. These final schedules are often compromises as it is often infeasible to match employee desires and skill levels with the call center requirements as determined in the previous stages. Employee scheduling and rostering are two very difficult problems that are still receiving much attention from OR/MS professionals and academics.

1.2 The Profile of an Individual Incoming Call

As an incoming call is processed by a call center, it may be designated by many different names, depending on its current state and the path by which it traveled. Figure 2 below helps describe the possible pathways and classification system. Calls are randomly *offered* to the call center and if there is still trunk line capacity, the call is *accepted* by a computer switch called an Automatic Call Distributor (ACD), otherwise, the call is *blocked* and the caller receives a busy signal. The ACD routes the calls to available TSRs. A very basic ACD will simply route calls to the next available operator but more modern ACDs are programmable and allow call-dependent call routings that may be controlled by TSRs, the telephone number called by the customer, Automatic Number Identification (ANI) or Interactive Voice Response (IVR).

• Figure 2: Incoming Call Nomenclature



The ACD not only routes calls, but also keeps track of how long calls have been in the system as well as other statistics. Often, when all available agents are busy, the caller will be presented with a recording ("*Your call is important to us...*") and asked to wait online. The caller may choose to wait patiently, or to *abandon* the call and perhaps try again later. Most ACDs will record abandonments – total number and how long the caller waited before hanging up (*time to abandonment*). These statistics are often provided to call center management in the form of a daily report as in Table 1 below.

• Table 1: Typical Abandonment Report from an ACD

Time Block	Calls Accepted	Calls Handled	Calls Abandoned (seconds)				
			0-30	30-60	60-120	120 – 240	240+
8:00 – 8:30 am	24	20	1	1	2	0	0
8:30 – 9:00 am	36	30	2	0	2	1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
3:30 – 4:00 pm	11	10	0	0	1	0	0

The ACD also keeps track of calls that remain in the system to be *handled*. It records the amount of time the call waits before being answered (*time to answer*) and the time the caller spends with an agent (*handle time*). These data are carefully recorded, but often not stored for more than one time block. At the end of each time block, the tally and times for individual calls are summarized; total # of calls entered, blocked, accepted, abandoned, handled during the time block, average handle time and average time to answer are printed to file. Individual call statistics generally do not appear to be kept in any retrievable form for later use. ACDs will often return data as seen in Table 2 below.

• Table 2: Typical ACD output (on left) & the individual call handle time data that is 'lost' (on right).

	Time block (j)	# Calls (n_j)	Average handle time ($\overline{X}_j$)
1	8:00 am – 8:30 am	20	6.2
2	8:30 am – 9:00 am	20	6.0
3	9:00 am – 9:30 am	30	5.5
4	9:30 am – 10:00 am	40	5.9
5	10:00 am – 10:30 am	40	5.8
6	10:30 am – 11:00 am	40	6.0
7	11:00 am – 11:30 am	50	5.9
8	11:30 am – 12:00 noon	50	5.9
9	12:00 noon – 12:30 pm	40	5.9
10	12:30 pm – 1:00 pm	40	6.4
11	1:00 pm – 1:30 pm	50	6.1
12	1:30 pm – 2:00 pm	40	6.5
13	2:00 pm – 2:30 pm	30	6.1
14	2:30 pm – 3:00 pm	20	6.6
15	3:00 pm – 3:30 pm	20	5.6
16	3:30 pm – 4:00 pm	10	5.8

Individual calls (X_i)
6.61
5.93
6.05
3.39
5.88
5.09
9.01
3.54
7.83
4.59

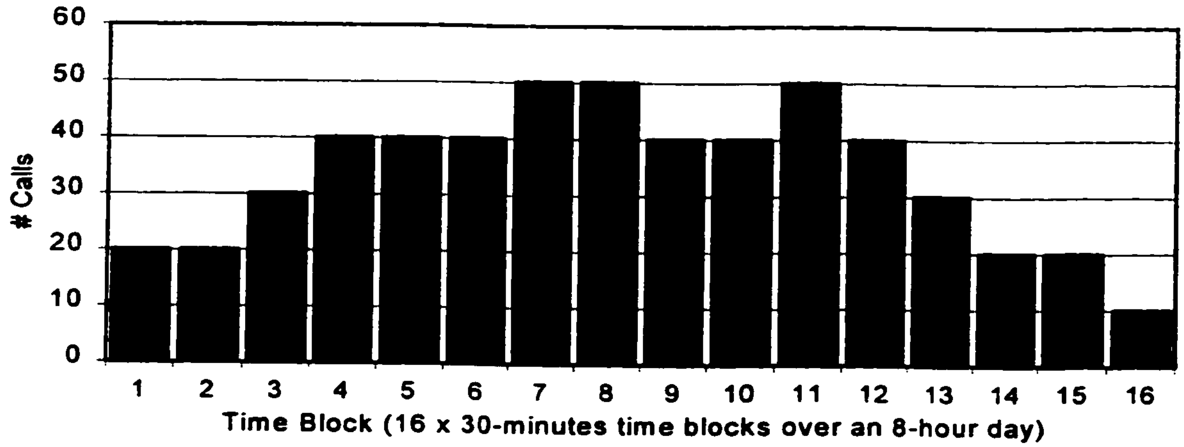
Where 5.8 is the average of these 10 individual calls above

$$\overline{X}_j = \frac{\sum_{i \in Block\ j} X_i}{n}$$

For each time block, the actual number of calls that arrived, and the average length of those calls are recorded. For most call center managers, this is more than enough information... and often this data is summarized even further into daily or weekly reports. For the analyst wishing to build an accurate call center simulation though, the 'good data' has been lost, deleted at the end of each time block and unrecoverable. These 'missing' individual call handle times are needed to accurately select the best distribution (and parameters) to model service time.

Average arrival rates for incoming calls can also be readily attained from these summary ACD reports. Unlike service times, where the average is inadequate, average arrival rates are sufficient for modeling arrivals. The analyst, assuming a Poisson arrival process, is primarily interested in the average number of calls that arrive over a given period of time. The 30 or 15-minute time blocks used for forecasting and scheduling often result in changing expected volumes from time block to time block. The forecast is composed of both a call pattern – often one distinct pattern for each day of the week, along with a call volume forecast. Figure 3 represents a sample call pattern that will be referred to throughout this paper. Each 30 minute time block receives 10 - 50 incoming calls. Example call volumes are multiples of ten for simplicity only.

• Figure 3: Sample Call Pattern with a Call Volume of 540 calls



In summary, while the call arrival rate (pattern) is readily attainable from ACD reports, the handle time (or call length, call duration) of individual calls has been replaced with averages. For most call center managers, average handle time has always been sufficient input into queuing theory tables/calculators. With more centers using computer simulations to estimate performance, analysts are no longer limited to the exponential service time assumption and **individual call handle times are needed to confirm or determine call handle time distribution and to build effective and accurate computer simulations.** These individual call lengths no longer exist -- they have been aggregated into a time block average. It will be discussed in the following sections how aggregation affects the information available to the analyst, how the aggregated data is currently being used by call center analysts, and why it is important to recapture critical information about the individual calls, through a process of disaggregation.

1.3 The Profile of Aggregated Calls

The problem of data aggregation can be demonstrated by graphing both aggregated and individual call handle times side-by-side. Synthetic ACD data (see section 4.1) was created by simulating three sets of randomly generated call handle times: exponential (μ=6), censored normal (μ=6,σ=2)[1], and lognormal (μ=6,σ=4). The data were then used to represent the individual call data 'lost' during aggregation.

In all 3 examples, 540 daily calls arriving over an eight-hour day were distributed across sixteen 30-minute time blocks according to a fixed call pattern. The arrival call pattern was that shown in Figure 3 where the number of calls arriving during each time block range from 10 to 50 calls. For each time block, the call handle times were averaged and both the aggregated and individual call handle times were plotted (Figure 4 - Figure 6). The data given in Table 2 shows the results of aggregating individual calls with a call handle time ~N(6,2) (censored). Exponential and lognormal call handle times were also generated using the same method (see Appendix B.2 and B.4).

An important observation is that each of the three generated samples (exponential, normal, lognormal) produce the same 'aggregated call handle time' histogram. That is, each of the three distributions were able to produce[2] the same histogram of aggregated call handle times: 300 calls were handled in time blocks with an average between 4 and 6 minutes and 240 calls were handled in time blocks with an average between 6 and 8 minutes. Unlike these examples, the data used to create the 'aggregated' histogram is *the only information* that the call center analyst has to make conclusions about individual call handle time. The following histogram overlays (Figure 4 - Figure 6) highlight the two critical

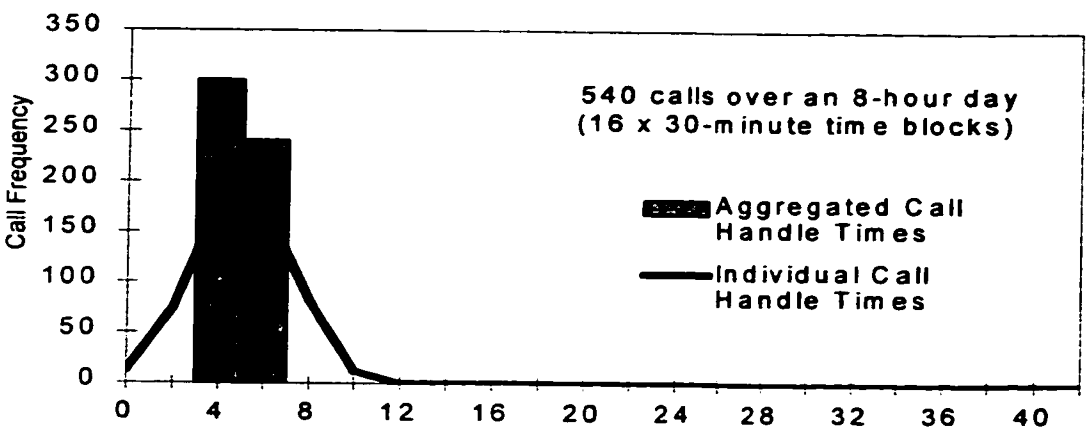
[1] Any negative call lengths generated by the Normal(6,2) distribution were given a call length of zero.
[2] Each of the three distributions were able to produce a range of aggregated data histograms -- several iterations were required to arrive at the same histogram across all three distributions.

concerns of using aggregated data to determine the distribution of individual calls:

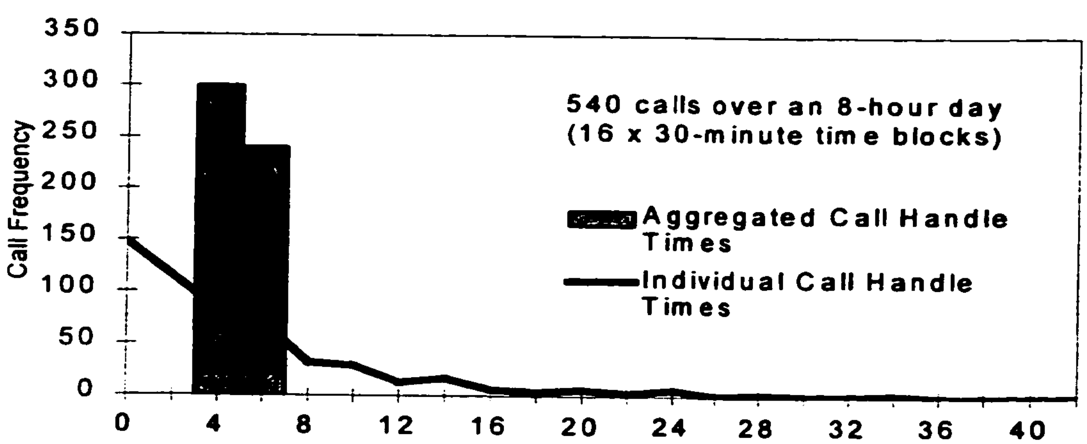
1. **The variance suggested by the aggregated data does not accurately reflect the variance of the individual calls. (as will be discussed in section 3.3)**

2. **The shape of the resulting histogram provides little clue to the distribution of individual call handle times.**

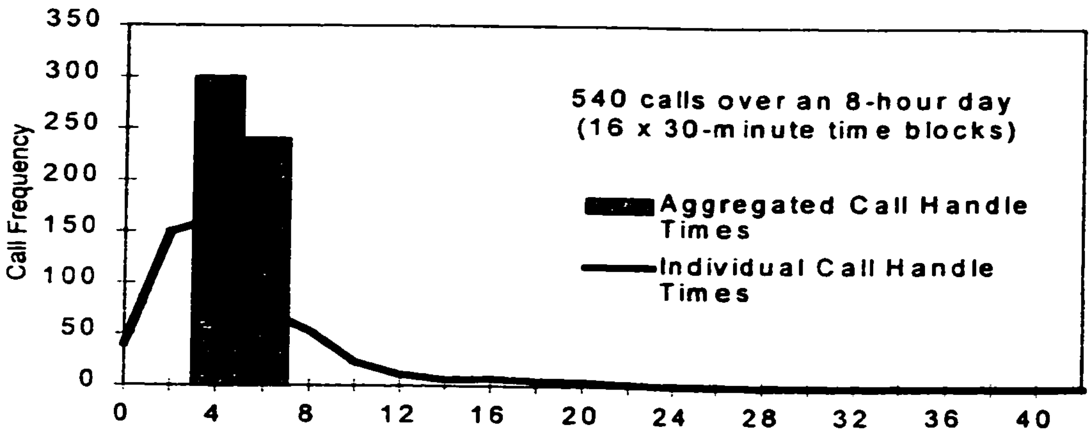
• Figure 4: Histograms of Normally Distributed [$\mu = 6, \sigma = 2$] Call handle time (Synthetic Data)



• Figure 5: Histograms of Exponentially Distributed [$\mu = 6, \sigma = 6$] Call handle time (Synthetic Data)



• Figure 6: Histograms of Lognormally Distributed [$\mu = 6$, $\sigma = 4$] Call handle time (Synthetic Data)



1.4 Current Practices in Call Center Modeling & Data Collection

In this section, current call center practices will be discussed in the same order that each issue would be addressed in a call center. That is:

1) How am I going to estimate the performance of my call center? Will I use queueing theory or will I need to invest in a more complex computer simulation?

2) I have 500 gigabytes of data from my ACD! How can I summarize it into something useful?

3) What assumptions am I going to make in order to build my model?

1.4.1 Current Methods of Estimating Performance: Queueing Theory vs. Simulation

Call centers vary greatly in their use of technology to help model call arrivals and service. While use of queueing theory seems to be predominate [Cleveland & Mayben 1997; Mehrotra 1997; Mehrotra et al 1997], there is an increasing use of computer simulation. Monte Carlo simulation is a useful tool when queueing theory assumptions no longer hold true. There are many reasons why queueing theory may not be the best choice in a modern call center:

- Queueing theory is often used as a 'black box' -- those using Erlang tables/calculators may not understand the origin and limitations of the methods.
- Queueing theory assumes that there are no abandonments, which is unrealistic for most call centers.
- Traffic intensity (demand for service divided by service capacity) cannot always be assumed to be less than 1.
- Steady-state behavior cannot be expected as time blocks become increasingly smaller, call volumes fluctuate and staffing levels are adjusted from one time block to another.
- Exponential service times with a mode of zero are questionable, especially since service times are often a convolution of both talk time and after call work.
- Complex call routings are becoming more common, and queueing theory is unable to model these routings adequately.

Given the above limitations, several industry experts and researchers indicate that under certain circumstances, simulation is preferred for modeling call centers [Mehrotra 1997; Mehrotra et al 1997; MacPherson 1988; Cleveland & Mayben 1997]. Once the modeling technique has been chosen, the next step is to collect and analyze the ACD data to determine how to model the input.

1.4.2 Data Modeling

Once the data has been gathered into an ACD report, the analyst must carefully decide how to interpret and use the data. When attempting to model the data received from the ACD, **a common (and accepted) method is to assume an exponential distribution with mean equal to the weighted-average of the call handle times**. This approach is based on the assumptions of classical queueing theory – specifically exponential service times.

Using the same method as described above and the data from Table 2, we can calculate the weighted-average of the (m =16) sample means as:

• Equation 1

$$\overline{\overline{X}} = \frac{\sum_{j=1}^{m} n_j \overline{X}_j}{\sum_{j=1}^{m} n_j} = \frac{(20 \times 6.2) + (20 \times 6.0) + \ldots + (20 \times 5.6) + (10 \times 5.8)}{20 + 20 + \ldots + 20 + 10} = 6.0$$

where m is the number of samples (time blocks)
n_j is the number of calls in each time block j
X_j is the average call length for each time block j
$\overline{\overline{X}}$ is the average call length (weighted) across m time blocks

Using the data in Table 2, an exponential distribution with a mean of 6 minutes would be used to model the call handle time. The exponential distribution has a single parameter (λ) that simultaneously describes the population mean ($1/\lambda$) and variance ($1/\lambda^2$). The sample mean in Equation 1 is an unbiased estimator of population mean and hence, the parameter λ can be estimated without estimating population variance. This is a distinct advantage of the exponential service time assumption and will be explored further in Section 3.3.

1.4.3 Incoming Call Data Assumptions

For the purpose of this paper, it will be assumed that **all calls throughout the day are of the same 'call type' – that is, they follow the same call handle time distribution during each time block**. This seems to be a standard assumption in the industry since most computer simulations and queueing theory methods are based on this assumption also. Some may find this assumption troublesome since it is reasonable that different time blocks will 'attract' a different call type and hence, a different call distribution.

Another common call center assumption that will be adopted is that the **ACD is recording data correctly**. This assumption may seem trivial, but there is some question about the time block in which calls are actually recorded. It shall be assumed that the call arrival and associated handle time are recorded in the same time block in which the call arrives, but there is evidence that many ACDs may actually attribute the call handle time to the time block in which it was completed. Violation of this assumption impacts not only input data analysis, but also forecasting, as call volumes may contain inaccuracies.

1.5 Definition of Research Problem

Simply stated, this paper will address the research problem, "How can the aggregated (averaged) data provided by most ACDs be used to parameterize and select the best theoretical distribution to model individual call handle times?"

An audience that should be interested in the results of this research are those involved with the stage of performance estimation, particularly when computer simulations are being used to anticipate service level. Many 'solutions' will be proposed — several of which are inadequate, a fact that will perhaps be immediately obvious to some readers. The reason these solutions will be discussed is that they are often used or attempted by inexperienced analysts (and students) new to call center simulation. A contribution of this research is to demonstrate the flaws in commonly used approaches.

The goal of this thesis is to demonstrate effective methods for arriving at the distribution of individual call handle times. This distribution is an essential component of an accurate simulation model, but whether using a computer simulation or not, this research may help provide insight into the distibution of the duration of incoming calls.

1.5.1 Objectives of Research

- Demonstrate how current methods of Input Data Analysis are inappropriate with aggregated ACD Data
- Identify methods of handling 'Aggregated Call handle time' data to arrive at usable input for a call center simulation.
- Demonstrate the accuracy and efficiency of these methods
- Make recommendations to practitioners with respect to the best method to use in various situations.

1.5.2 Scope of Research

- It will be assumed that actual call handle times follow either an exponential, a censored normal (since call handle times > 0), or a lognormal distribution.
- A model call center with a moderate daily call volume of 540 calls spread unevenly over an eight hour day (16 x 30 – minute time blocks) will be used to demonstrate the effects of aggregation and the proposed 'disaggregation' methods. The methods used could be applied to other call center configurations.
- The call center will handle only the one type of call by a single tier of TSRs. When a call center has a variety of call types, a separate analysis of each would be appropriate.
- The data analysis will be limited to synthetic data created to test potential disaggregation methods. The absence of 'real' data highlights the difficulty analysts have obtaining individual call lengths.

1.5.3 Application and Relevance of Research

As long as call centers continue to use ACDs that handle data in the fashion described, there will be a need for effective disaggregation methods. As mentioned previously, the majority of call centers continue to use queuing theory (often in the form of Erlang tables), and if simulations are used, exponential service times are often an automatic assumption. Until those analyzing call center performance begin to make use of simulation tools, question service time assumptions and demand more useful data from their ACDs, the effects of temporally aggregated service time data will require special treatment from analysts.

Chapter 2 Literature Review

Little, if any, has been written specifically about developing methods to disaggregate data for the purpose of determining input models for discrete event simulation. The following literature survey touches upon many related topics in order to provide a basis for comparison and potential insights. There has been a great deal written lately concerning Bayesian Inference and the use of Markov chain Monte Carlo methods such as Gibbs Sampling. That literature will not be discussed in this section, rather it will be referred to extensively in section 3.4 when Bayesian inference is introduced.

2.1 Service times

Exponential service times have been a historical, but not proven, model for call handle times. Its use in queueing theory applications has continued in computer call center simulations with little comment on its relation to actual call handle time distributions. The teletraffic industry also uses exponential distributions to model telephone circuit holding time distributions, but there has been some discussion as to the validity of this assumption.

Bolotin (1994) provides empirical data showing that the conversation time variance is much larger than in the exponential distribution and that for an individual subscriber, the distribution call duration is lognormal.

Bolotin's reasons for choosing the lognormal distribution are compelling. He suggests that the callers' perception of time is on a logarithmic scale -- the longer a conversations is, the longer the call is likely to continue. For example, a caller that has already been in conversation for 1 minute will perceive the next 15 seconds of the conversation in a similar fashion as a caller that has spoken for 20 minutes will perceive the following 5 minutes of conversation.

i.e. value of time = log(15sec/60sec) = log (5min/20min) = log(0.25)

Bolotin proposes that mixtures of lognormal distributions characterize call duration for groups of subscribers, and demonstrates that a mixture of two lognormal distributions fits empirical data well. Others propose that the Weibull distribution (Rahko 1991) or the gamma distribution (Takemori et al. 1985) are also more appropriate distributions to model conversation times.

Additional research, specifically related to the call handle times at inbound call centers, is required to verify this teletraffic behaviour and determine an accurate call length distribution for modern call centers.

2.2 Aggregation of Data

Data can be aggregated temporally, geographically, by market level or according to other classification systems. For example, time aggregated data is often a problem for economists who wish to prepare monthly forecasts when only quarterly historical data is available. Marketing analysts often encounter data that has been aggregated geographically or by market level, and may wish to make projections based on sub-regions. Inventory managers may be presented with forecasts for product families, but instead require demand information about components. These are just a few examples of circumstances in which aggregated data must somehow be 'disaggregated' in order perform further analysis.

Economists are often faced with data series of different temporal aggregation. Chow and Lin (1976) developed a disaggregation method (best linear unbiased method) that is still often used (Abeysinghe and Lee 1998). Once the disaggregation has been completed, these values may be inserted into the model and treated as actual observations. Hsiao (1979) presents a maximum likelihood approach that performs data prediction and parameter estimation simultaneously, although its solution is computationally complex (Palm and Nijman 1982).

Aggregated data can also be used to update our beliefs about its constituent parts. There are often many circumstances where information is only known about the sum of random variables and given that sum, we wish to modify our prior belief about the distribution of the individual components. This is the case presented by Jonsson and Silver (1987). The authors present a similar approach to two situations: one where the sum of demands at several locations is known exactly and another where the sum is known to be less than a given value. In these two cases, the authors use a Bayesian paradigm and considerable algebraic manipulation of conditional probabilities to arrive at an expression of the conditional density function for any one of the individual locations.

2.3 Bayesian Methods Applied to Stochastic Simulations

One can find many instances in the literature where Bayesian statistics and Monte Carlo simulation are discussed together. The use of simulation as a tool for implementing Bayesian analysis has been well documented over the past decade and will be discussed further in section 3.4. A more recent marriage of the two topics has only begun to receive attention and promises to bring interesting, new approaches to simulations.

Classical techniques for input distribution simulation (Banks and Carson 1996) involve selecting a single distribution model and point estimate(s) of accompanying parameter(s). The computer simulation is then performed and output analysis completed in order to determine the *stochastic* or *simulation* uncertainty (variance) for the given input. Recently, there has been much discussion of incorporating another source of uncertainty -- distribution selection and parameterization, often called *parameter, structural, systemic* or *subjective uncertainty* (Draper 1995; Chick 1997a,1998; Cheng and Holland 1997) into the simulation. These authors suggest that by taking a Bayesian approach, conditioning on model and parameter choice, and then integrating over the uncertainty in both, one can achieve more accurate results.

While these ideas are thought-provoking and eloquent, it will probably be many years before they are incorporated into call center simulation packages. In the mean time, simple sensitivity analyses incorporating model and parameter uncertainty should be encouraged.

2.4 Bayesian Methods Applied to Operations Management

As Bayesian statistics has received more attention from statisticians, practitioners from operations management have also begun to apply its methods to business problems. While the list below is far from exhaustive, it provides a sample range of applications.

- Ahn and Ezawa (1997) develop a Bayesian network learning model as part of a decision support system to help outbound call center TSRs determine an optimal customer service approach based on predicted response probability.
- Kaplan (1988) uses Bayesian updating to improve forecast demand in a dynamic programming formulation of a periodic review (s, S) inventory control model.
- McGrath et al (1987) present a Bayesian approach to queues that incorporates uncertainty about interarrival and service times. The authors recognize that certain queueing constraints, such as stability, can be directly accounted for by conditioning on stability.
- Silver and Fiechter (1995) develop an optimal Bayesian strategy to determine the preventive maintenance interval when the operating time distribution is not exactly known due to limited historical data.

Chapter 3 Development of Models

3.1 Disaggregation

Throughout this thesis, the term 'disaggregation' is used to describe a broad range of data manipulations. The purpose of these manipulations is to use the given aggregated (averaged) call handle times to arrive at an accurate description of call handle time distribution. These call handle time distributions can then be used as input into computer simulations to accurately model call center performance. In the following sections, 3 alternate approaches to modeling data similar to that found in Table 2 will be discussed. When the approaches are shown to be inadequate or incomplete predictors of the individual call handle time distribution, they are presented not to mislead, but rather as examples of methods that are often used by inexperienced analysts and students.

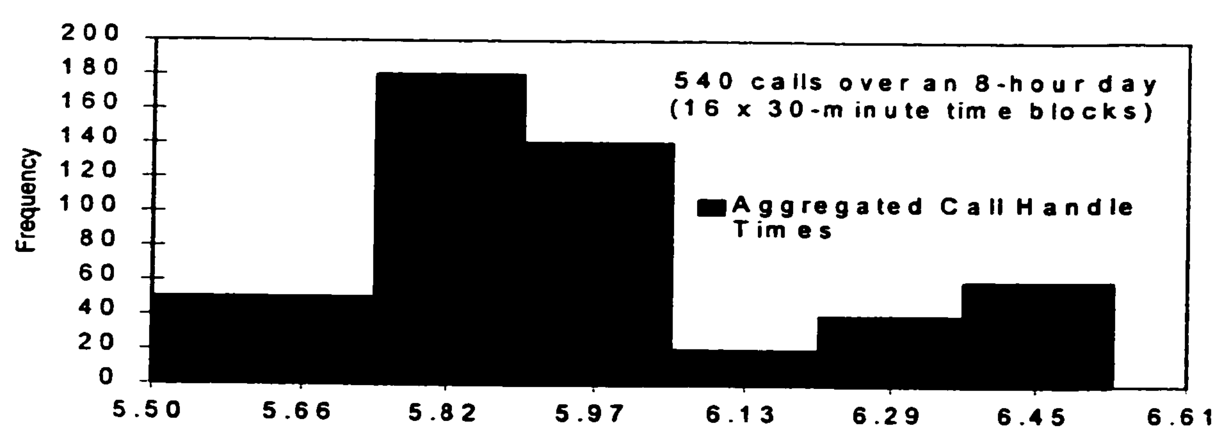
3.2 Standard Input Analysis Applied to Time-aggregated Data

Banks and Carson [1984] list 4 steps in the development of a valid model of input data. The first step is the collection of the raw data and the second step is the identification of the underlying statistical distribution, beginning with the creation of a histogram from the data available. From this frequency distribution (histogram), a *theoretical distribution* (normal, exponential, lognormal, beta etc.) that best represents the distribution of the observed random variable can be inferred. In the third step, estimates are made of the parameters that characterize the distribution. In the final step, a variety of test statistics (Chi-squared, Kolmogorov-Smirnov) are calculated in order to determine how well the theoretical distribution 'fits'. Alternatively, if none of the theoretical distributions can be used to model the data, the analyst then has the option of sampling directly from the historical data (a method often referred to as '*bootstrapping*') or creating a continuous empirical distribution ('*smoothed bootstrapping*') from which to sample (Cheng 1994).

The first model developed disaggregates the data in a very simplistic way. Analysts will often use a method similar to the 4-steps described above and for that reason, they would likely not assume an exponential distribution. These analysts often have a variety of statistical software tools that they use to help analyze data and arrive at distributions that best fit the data given to them. These data-fitting tools are widely available and are often packaged with simulation software.

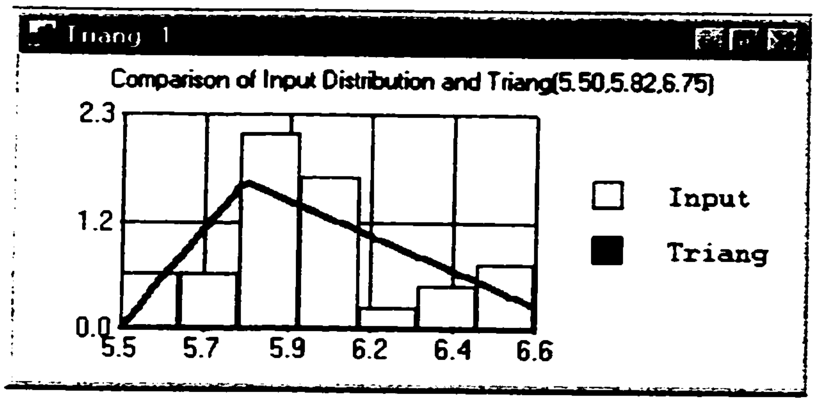
Using the data given in Table 2 as an example, the analyst could interpret the aggregated data as such: 20 observations of 6.2 minutes each, 20 observations of 6.0 minutes, 30 observations of 5.5 minutes, 40 calls of length 5.9 minutes etc. In a very basic sense, the analyst has disaggregated the 16 half-hourly observations into 540 individual observations. These observations can then be placed into a histogram with the hope of getting a better understanding of the frequency distribution. The histogram in Figure 7 below represents the data in Table 2 treated in such a manner..

• Figure 7: Histogram of Aggregated ACD Data given in Table 2

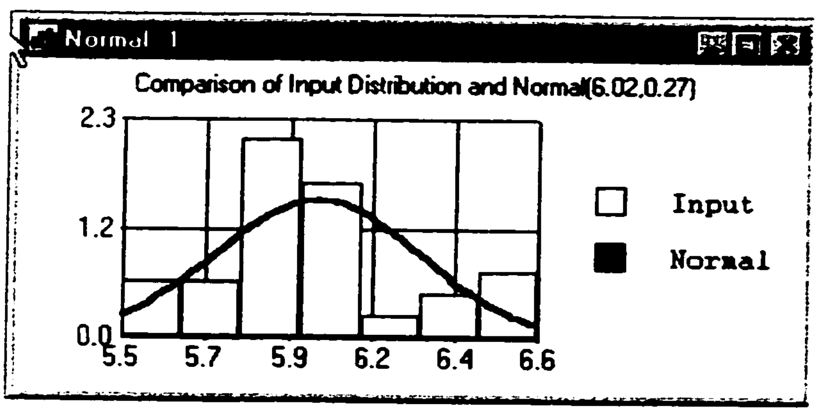


Once this data is in a histogram, the analyst may be able to choose a distribution from visual inspection or from use of a data-fitting software application. When the data from Table 2 is analyzed using a package such as Palisade's BESTFIT (Figure 8 - Figure 10), the distribution that provides the 'best fit' is either triangular, normal or beta, depending on the test statistic used. All three of the proposed models fail to achieve 'significance' according to any of the tests (Chi-Squared, Kolmogorov-Smirnov, or Anderson-Darling) since the data is too 'chunky' given the high number of observations. This will often be the case with any set of aggregated data treated in this manner.

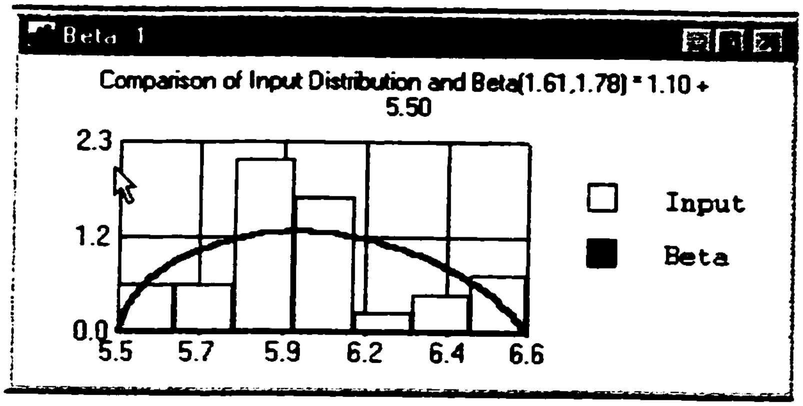
- Figure 8: Triangular Distribution Fit to Aggregated Handle Time Histogram



- Figure 9: Normal Distribution Fit to Aggregated Handle Time Histogram



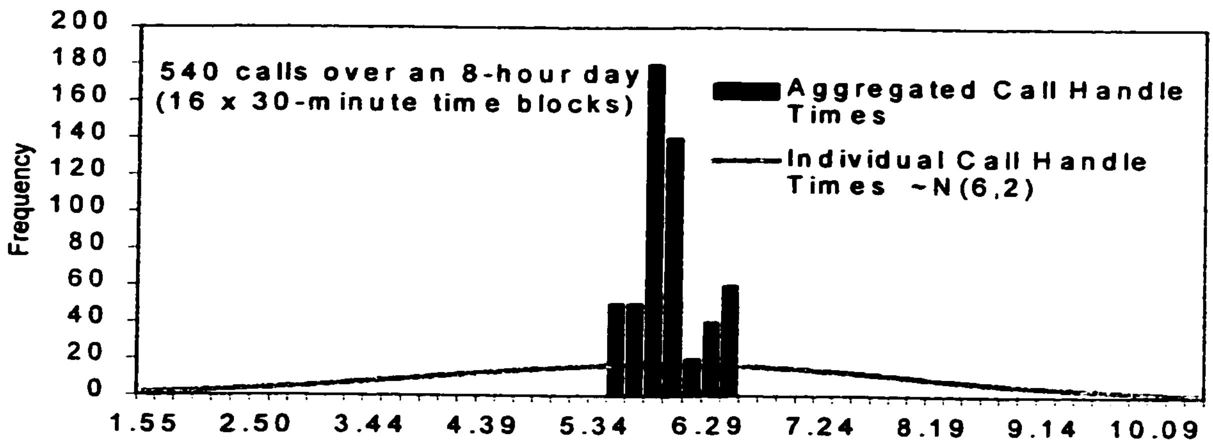
• Figure 10: Beta Distribution Fit to Aggregated Handle Time Histogram



Given that none of the theoretical distributions meets required critical values for any curve-fitting test statistic, which of the three above-mentioned distributions should the analyst use to model individual call handle times? The answer is probably NONE, since the average call handle times significantly understate the variability of individual calls. Each of these distributions would most likely perform adequately if the desired random variable was an average of 10 – 50 call lengths, but an effective call center simulation requires individual call handle times.

Since Table 2 contains simulated, synthetic data, we already know that the underlying distribution of individual calls is a normal random variate with mean equal to 6.0 and standard deviation of 2.0, censored at zero. When the frequency distribution is plotted with the histogram of aggregated data (as in Figure 11), it can be seen that the variance within the aggregated data is much less than the variance between individual calls. Hence none of the above distributions fitted to the aggregated data will adequately model *individual* call handle times. Each of the proposed distributions is centered on the population mean, but an accurate portrayal of the variance and overall shape of the individual call handle time distribution is not achieved. **In general, the standard input analysis methods fail to determine individual handle time distributions when presented only with aggregated data.**

• Figure 11: Histogram of Aggregated Data and Frequency Distribution of Individual Calls



3.3 Determining Distribution Parameters with Unbiased Estimates (UE)

3.3.1 Sample and Population Mean

In section 1.4.2, an unbiased estimator of population mean was given:

• Equation 1(repeated)

$$\bar{\bar{X}} = \frac{\sum_{j=1}^{m} n_j \bar{X}_j}{\sum_{j=1}^{m} n_j} = \frac{(20\times6.2)+(20\times6.0)+...+(20\times5.6)+(10\times5.8)}{20+20+...+20+10} = 6.0$$

3.3.2 Variance of Sample Averages

A mistake often made by inexperienced analysts is to then use the variance of sample averages, $s_{\bar{x}}^2$, as an estimate of population variance, σ^2. In this situation, $s_{\bar{x}}^2$ represents the variance within the aggregated data, while σ^2 represents the variance within individual call handle times. Unlike sample variance, s^2, $s_{\bar{x}}^2$ is not an unbiased estimate of σ^2. Two example calculations of $s_{\bar{x}}^2$ (Equation 2, Equation 3) for the data in Table 2 has been provided. Equation 2, while easily calculated and often mistakenly used, does not actually represent any statistical value, since it does not reflect that each sample contains a different number of observed individual calls. Equation 3 correctly calculates $s_{\bar{x}}^2$.

• Equation 2

$$s_{\bar{x}}^2 \neq \frac{\sum_{j=1}^{m}(\bar{X}_j - \bar{\bar{X}})^2}{m-1} = \frac{(6.2-6.0)^2 + (6.0-6.0)^2 + ... + (5.8-6.0)^2}{15} = 0.31$$

• Equation 3

$$s_{\bar{x}}^2 = \frac{\sum_{j=1}^{m} n_j(\bar{X}_j - \bar{\bar{X}})^2}{\sum_{j=1}^{m} n_j} = \frac{20(6.2-6.0)^2 + 20(6.0-6.0)^2 + ... + 10(5.8-6.0)^2}{539} = 0.28$$

The reader will recall that the individual call distribution was a censored normal distribution with a mean of 6 and a standard deviation of 2 or a variance of 4. The source of the downward bias is the fact that each of the calculated $\bar{X}_j$ is an *average* talk time for the specified time block and deviates from the mean significantly less than the unobserved individual talk times. This bias is quantified for large n and the population variance (individual call handle time variance) can be estimated using the relationship:

• Equation 4

$$s_{\bar{X}}^2 \approx \frac{\sigma^2}{n} \quad \textit{where} \quad n = \textit{the number of observations summarized by } \bar{X}$$

Analysts that recognize the downward bias present in the variance of sample averages are likely to try to 'adjust' for it using the above relationship. Unfortunately, σ^2 cannot be estimated using $s_{\bar{x}}^2$ and Equation 4 for two reasons:

- n differs from time block to time block and there is not one single 'factor' by which to multiply the sample variance.
- there may be time blocks in which n is small and Equation 4 will not hold.

As mentioned previously, s^2 is an unbiased estimator of σ^2, but ACDs generally do not record sample variance and without the individual call lengths, s^2 cannot be calculated. The inability to calculate the sample variance and subsequently estimate population variance often leads the analyst to adopt the assumption of an exponential service time. The exponential distribution has a single parameter (λ) that simultaneously describes the population mean ($1/\lambda$) and variance ($1/\lambda^2$) and hence, the parameter λ can be calculated with an estimate of population mean alone. The estimation of population variance is unnecessary and this is a distinct advantage of the exponential service time assumption.

For multi-parameter distributions such as the Gaussian (Normal), Lognormal or Gamma distribution, an unbiased estimate of population variance is required in order to determine the parameters of the distribution.

3.3.3 Population Variance

The search for a known, unbiased estimate of population variance for aggregated data led to the sum of squares formulas used in an ANOVA with unbalanced data (Guttman et al., p 387-388). In analysis of variance, the total variation, SS_T is broken down into two components: SS_B (between samples sum of squares), which reflects the variance between treatment groups and SS_W (within samples sum of squares), which reflects variation with treatments.

• Equation 5

$$SS_{total} = SS_{between} + SS_{within} \quad \textit{or alternatively,} \quad SS_T = SS_B + SS_W$$

The between sample sum of squares is calculated based on the difference between sample means and the overall mean (see numerator, Equation 7). The expected between samples sum of squares, $E(SS_B)$, is comprised of both population variance and variance due to the different 'treatments' of each sample, as shown in Equation 6.

• Equation 6

$$E(SS_B) = \sigma^2 (m-1) + \sum_{j=1}^{m} n_j \delta_j^2$$

where δ_j *are the effects of each time block* j

Assuming that the call handle time distribution is independent of the time block in which the call arrived (i.e. there are <u>no</u> treatment effects), each of the δ_j are equal to zero. Equation 6 can be simplified and rearranged to arrive at an unbiased estimate of the population variance, σ^2:

• Equation 7

$$\sigma^2 \approx \frac{SS_B}{m-1} = \frac{\sum_{j=1}^{m} n_j (\overline{X}_j - \overline{\overline{X}})^2}{m-1} = \frac{\sum_{j=1}^{m} \overline{X}_j^2 n_j - \overline{\overline{X}}^2 \sum_{j=1}^{m} n_j}{m-1}$$

Using Equation 7 and the data from Table 2, we can calculate the unbiased estimate for population variance as done below:

• Equation 8

$$\sigma^2 \approx \frac{SS_B}{m-1} = \frac{\sum_{j=1}^{m} n_j (\overline{X}_j - \overline{\overline{X}})^2}{m-1} = \frac{20(6.2-6.0)^2 + 20(6.0-6.0)^2 + \cdots + 10(5.8-6.0)^2}{15} = 2.8$$

For the data in Table 2, the individual calls were censored normal random variates with a mean of 6.0 and a variance of 4.0. The above estimate of population variance of 2.8 should not be viewed as a poor estimate since the statistic $\frac{SS_B}{m-1}$ is a random variable itself, which has its own distribution. The mean of this distribution is σ^2. Increased accuracy of the variance estimate may be achieved by sampling over a greater number of time periods rather than just the 16 time blocks used in this example.

Once estimates of population variance and mean have been calculated, the next step is to determine which distribution to use and to calculate the distribution parameters. For many theoretical distributions, calculation of the parameters is quite simple once population mean and variance have been estimated. The difficult task is choosing which distribution to use! As seen in the previous section, the aggregated data provides little clue as to the underlying distribution of individual calls.

The coefficient of variation (c.v.) is defined as the ratio of standard deviation to mean ($\frac{\sigma}{\mu}$). As a simple rule of thumb, if $c.v. \cong 1.0$, then an exponential distribution may be appropriate. Otherwise, if $c.v. < 1.0$, then a normal distribution may be required. Several other distributions, such as Gamma or Weibull may fit just as well, or perhaps better. Unfortunately, one cannot evaluate distributions using a trial and error method by comparing test statistics (Chi-square, Kolmogorov-Smirnov) since the test statistic is calculated based on the observed call handle time data, of which there is none!

The ability to calculate an unbiased estimate of population variance is a first step at understanding the distribution of individual call lengths, but without individual handle time data, **the unbiased estimates of variance and mean do not allow us to determine the best theoretical distribution for individual calls.** In order to use these estimates, more data would need to be collected about the individual calls in order to perform tests for goodness-of-fit.

3.4 A Bayesian Approach to Disaggregation

The third (and final) approach to be discussed is the use of Bayesian Inference to determine the best distribution and parameters to model individual call handle times. Before discussing the actual models, it would probably be helpful to briefly describe Bayesian Inference and Markov chain Monte Carlo (MCMC) methods as they apply to the models presented.

3.4.1 An Introduction to Bayesian Inference

In general terms, Bayesian inference often deals with circumstances where there is observed data, Y, and unknowns θ which may include model parameters, missing data, or events that have not been directly observed. Both Y and θ may be considered vector quantities.

It is assumed that from probability density functions the *likelihood, $p(Y|\theta)$*, can be determined. There is generally uncertainty about θ before analyzing the data; this is captured in the *prior distribution, $p(\theta)$*. Bayes Theorem provides a method of modifying the prior distribution based on the observed data. The end result is a *posterior distribution, $p(\theta|Y)$* that better represents the uncertainty in the unknown parameters, θ.

joint probability distribution

• Equation 9: Bayes theorem

$$p(\theta \mid Y) = \frac{p(\theta, Y)}{p(Y)} = \frac{p(\theta)p(Y \mid \theta)}{p(Y)} = \frac{p(\theta)p(Y \mid \theta)}{\int p(\theta)p(Y \mid \theta)d\theta} \propto \underbrace{p(\theta)p(Y \mid \theta)}_{\text{unnormalized posterior distribution}}$$

posterior distribution *prior distribution* *likelihood*

unnormalized posterior distribution

Since the denominator is independent of θ, it can be considered constant for a fixed Y and omitted to yield the *unnormalized posterior density*, the far right hand side of Equation 9.

Equation 9 demonstrates how the posterior distribution is simply a function of the prior distribution multiplied by the likelihood of the observations given the prior distribution. Obviously, the choice of prior becomes quite important, but where does the prior come from? Despite the considerable research into this topic, there is still no definitive or best way to choose a prior.

In principle, the choice of prior is subjective. The prior may attempt to capture existing data through a meta-analysis, or it may be determined with the input from 'experts'. Often classical statisticians will cite this subjectivity as a major weakness in Bayesian inference, and non-informative, 'reference', priors are often used to increase the objectivity of the analysis. The selection of prior distributions by formal rules is a goal of many statisticians, but there are still no definitive standards (Kass & Wasserman 1996).

Sensitivity analysis can be performed on a variety of priors as a form of validating the rigour of the model. Spiegelhalter [1999] advocates the use of a 'community of priors':

- reference priors representing non-informative, objective prior beliefs
- clinical priors that represent genuine clinical opinion
- archetypal priors expressing skepticism or enthusiasm

By performing a thorough sensitivity analysis using several priors, one can better assess how convincing the results will be to a broad spectrum of opinion.

Priors are also often chosen for mathematical convenience and are based on the form of the likelihood function. When the posterior distribution follows the same parametric form as the prior distribution, then the prior is said to be a *conjugate prior*. For example, if the likelihood follows a binomial distribution, then a beta prior is often used since the resulting posterior distribution will also be beta. The beta prior distribution is a *conjugate family* for the binomial likelihood. Other examples are shown in Table 3. As computer numerical methods have gained widespread use, restricting oneself to conjugate priors has not been as critical as it once was.

• Table 3: Conjugate families for several likelihood distributions.

Prior Distribution	Likelihood	Posterior Distribution
Normal	Normal (known variance)	Normal
Beta	Binomial	Beta
Dirichlet	Multinomial	Dirichlet
Gamma	Poisson	Gamma

3.4.1.1 Example of Bayesian Inference

Part A: Using a discrete prior distribution

Assume a new campaign at an outbound call center may have a success rate, θ, of 10%, 25%, 40%, 55% or 70%, each of equal prior probability ($p(\theta_i) = 0.2$). If we observe r successes out of n calls, how is our belief revised?

The likelihood function is binomial;

$$p(r \mid \theta) = Bin(r \mid n, \theta) = \binom{n}{r} \theta^r (1-\theta)^{n-r} \quad \propto \quad \theta^r (1-\theta)^{n-r}.$$

If r= 18 successes and n=30 calls, then the posterior probabilities may be calculated as in Table 4 below.

• Table 4: Calculation of Posterior distribution based on discrete prior

i	θ_i	Prior $p(\theta_i)$	Likelihood x Prior $\binom{30}{18}\theta_i^r(1-\theta_i)^{n-r}p(\theta_i)$	Posterior $p(\theta_i\|r)$
1	0.10	0.20	${}_{30}C_{18}\,(0.10)^{18}(0.90)^{12}(0.20)= 0.0000$	0.00
2	0.25	0.20	${}_{30}C_{18}\,(0.25)^{18}(0.75)^{12}(0.20)= 0.0000$	0.00
3	0.40	0.20	${}_{30}C_{18}\,(0.40)^{18}(0.60)^{12}(0.20)= 0.0026$	0.06
4	0.55	0.20	${}_{30}C_{18}\,(0.55)^{18}(0.45)^{12}(0.20)= 0.0253$	0.59
5	0.70	0.20	${}_{30}C_{18}\,(0.70)^{18}(0.30)^{12}(0.20)= 0.0150$	0.35
Σ_i		1	$p(r=18,n=30) = 0.0429$	1.00

As can be seen in column 4, the ${}_{30}C_{18}$ term is independent of θ, and can be considered a constant for a given n and r. ∴ it is common practice to omit this term from the calculation.

We now believe there is only a small chance of the campaign having a success rate of less than 40% and that the probability of the rate being 55% is ten times greater than 40%.

Part B: Using a continuous prior distribution.

What if instead of using a discrete prior as in part A, we had decided to use a continuous prior where we believed the success rate to have a mean of 0.40 and a variance of 0.015?

Since the likelihood function is still binomial as before, we choose a conjugate prior following the beta distribution.

This term may be considered constant – it is independent of θ

$$p(\theta) = Beta(\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1} \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

The posterior distribution can then be calculated as follows:

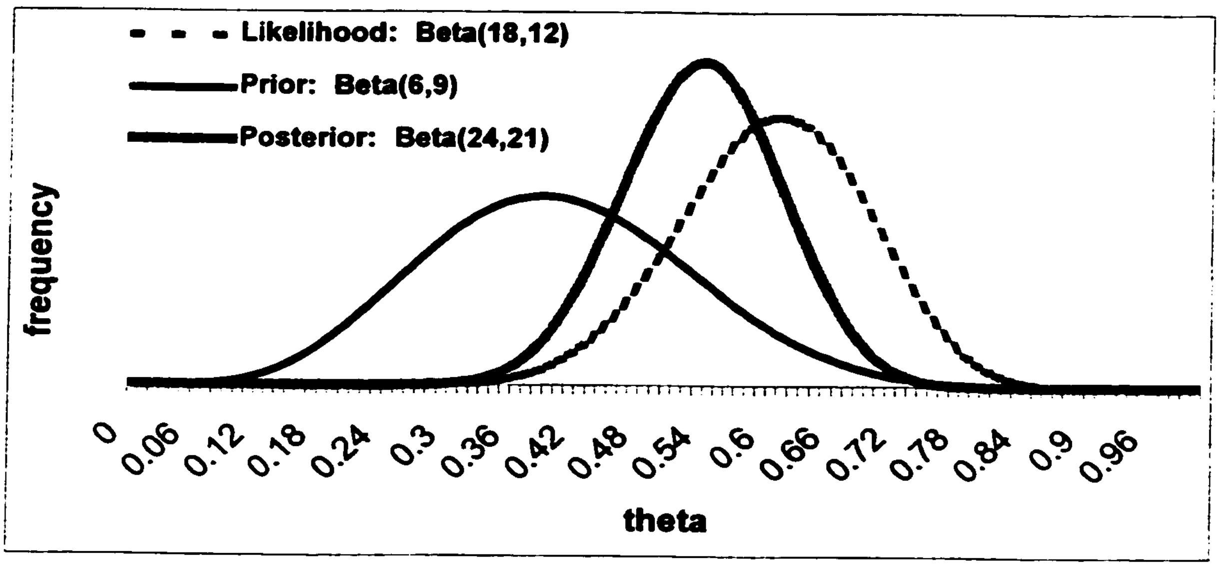
$$\begin{aligned} p(\theta \mid r,n) &\propto p(r \mid \theta,n)p(\theta) \\ &\equiv \theta^{r}(1-\theta)^{n-r}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &\equiv \theta^{r+\alpha-1}(1-\theta)^{n-r+\beta-1} \\ &\propto Beta(r+\alpha,n-r+\beta) \end{aligned}$$

•

The Beta distribution has mean $\mu = \alpha/(\alpha+\beta)$ and variance $s^2 = \mu(1-\mu)/(\alpha+\beta+1)$. With the prior having a mean = 0.40 and s^2=0.015, we can solve for α and β to find α= 6 and β= 9.

The posterior distribution for θ is then Beta(24,21) with a mean = 0.53 and s^2 = 0.0054. The binomial likelihood where r = 18 and n = 30 is proportional to a Beta(18,12) density function. We can graph all three together as seen in Figure 12 below to show visually the relationship between prior, likelihood and posterior distributions.

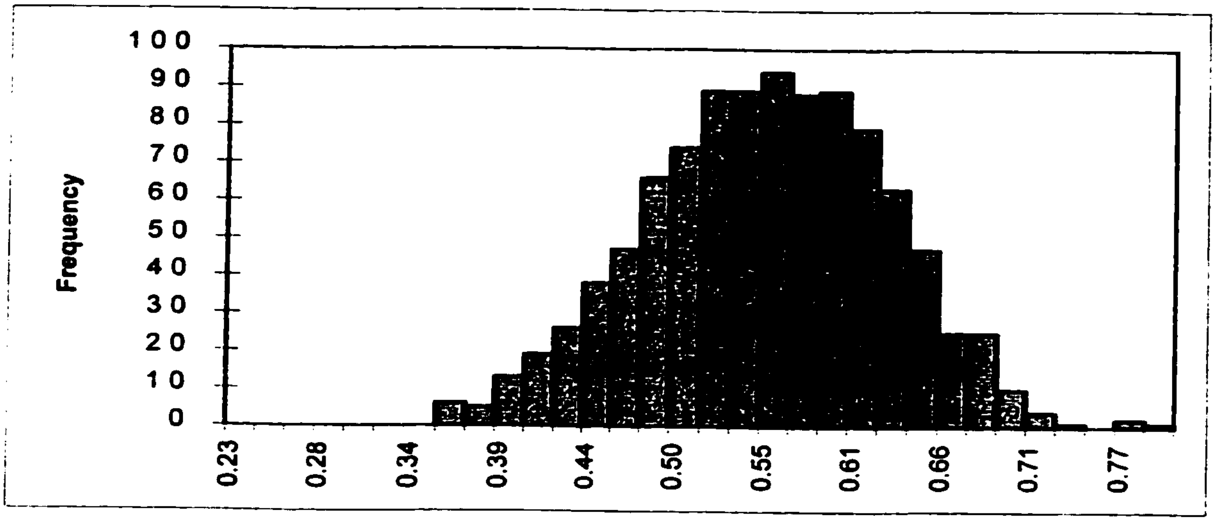
• Figure 12: Frequency Distributions of Prior, Likelihood and Posterior for example, part B



Unlike this simple example, it is often not feasible to perform calculations on the posterior density function directly, in which case it is useful to simulate from the posterior distribution to obtain inferences. This method is often used for the more complex models involving multiple variables of interest.

To illustrate this method, Figure 13 shows a histogram of 1000 draws from the posterior distribution. An estimate of the 95% posterior interval can be obtained by taking the 25th and the 975th of the 1000 ordered draws. In this case, an estimate of the 95% posterior interval is [0.395,0.678]. Unlike more complex posterior distributions, we can compare these values with those obtained from the known cumulative distribution function (cdf) for Beta variables. The 95% interval obtained from the BETAINV function in EXCEL is [0.388, 0.675].

• Figure 13: Histogram of 1000 draws from the posterior distribution: Beta(24,21)



3.4.2 Moving from Single Parameter to Hierarchical Models

Both of the above examples represent single parameter models in which inferences are made about a single unknown variable -- in this case, θ, the success rate of the new campaign. For most practical problems in statistics though, there is often more than one unobservable quantity and it is in these situations that the strengths of Bayesian inference become more pronounced.

When two or more variables are unknown, we are often not particularly interested in all of them. For example, if θ is a vector, $\theta = (\theta_1, \theta_2)$ where $\theta_1 = \mu$ and $\theta_2 = \sigma^2$ and we are primarily interested in θ_1 (μ) for the moment, then θ_2 (σ^2) is called a *nuisance parameter.* This label may seem a bit harsh, but if we are interested in the posterior distribution, $p(\theta_1|y)$, then the unknown variable θ_2 does become a bit of a nuisance from a calculation standpoint.

The initial step is to determine the *joint posterior distribution* of all unknowns:

• Equation 10

$$p(\theta \mid Y) = p(\theta_1, \theta_2 \mid Y) \propto p(Y \mid \theta_1, \theta_2) p(\theta_1, \theta_2)$$

Since the desired probability is the marginal probability of θ_1, we need to 'average over' or 'integrate out' θ_2:

Marginal probability of θ_1

• Equation 11

$$p(\theta_1 \mid Y) = \int p(\theta_1, \theta_2 \mid Y) d\theta_2 = \int p(\theta_1 \mid \theta_2, Y) p(\theta_2 \mid Y) d\theta_2$$

For multiparameter models, we rarely evaluate the integral in Equation 11 explicitly since it is often intractable for most real-scale problems. Rather, we simulate the factored form (far right side) by repeatedly sampling θ_2 from its marginal posterior distribution, and then sample θ_1 from its conditional posterior distribution, using the θ_2 drawn in the first step. The integration in Equation 11 is thus performed indirectly through simulation and is often called Monte Carlo Integration.

There are many other ways of approximating integrals, but for high-dimensional hierarchical models, Markov chain Monte Carlo methods (see section 3.4.3) have proven extremely useful (Evans and Swartz 1995). A *hierarchical model* is a multiparameter model in which the parameters of the prior distribution are themselves modeled as random variables following a specific theoretical distribution with its own parameters, called *hyperparameters*.

For example, let us assume we are to build a Bayesian model to make inferences about the parameter θ, given observed data Y. Rather than modeling the prior distribution of θ as a normal distribution with a mean of say, 10 and a standard deviation of 2, a more flexible prior can be used. In a hierarchical model where non-informative priors are used, a possible modeling of θ is:

$\theta \sim N(\mu, \tau)$[3] where $\mu \sim$ Normal (0,0.001)

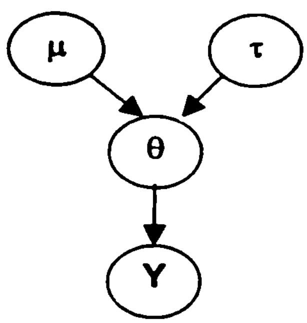
This produces a diffuse, non-informative prior centered at zero with $\sigma^2 = 1000$ (see footnote 3 below)

and precision, $\tau \sim$ Gamma(0.001,0.001)

This also produces a diffuse non-informative prior restricted to be positive.

Hierarchical models are often displayed graphically to help describe the essential structure of the model, to communicate conditional dependencies, to break the model down into simple components and to provide a basis for later simulation.

• Figure 14: Graphical representation of a Hierarchical Bayesian Model



Hierarchical models are obviously more complex than single parameter models, so why would we prefer to model in such a way? The answer is that many real-life systems involve stochastic variables that should be thought of as related, and their parameters should be allowed to reflect the dependence amongst these variables.

[3] In order to maintain consistency with the modeling language that will be described shortly, variance will be described in terms of its multiplicative inverse, *precision*. The symbol, tau (τ), is used to represent precision where $\tau = \frac{1}{\sigma^2}$ or $\sigma = \sqrt{\tau}^{-1}$. The normal distribution will then be parameterized as N(μ,τ) rather than N(μ,σ).

Expanding our example, let us assume that the observed data represents many measurements of a similar nature -- for example, the number of successes and trials at each of four call centers. Rarely will each measurement lead to exactly the same estimate of θ. Since each of the $p(\theta_j)$ will differ, our ability to predict Y will be hindered. If we are willing to assume that each of the θ are similar, then we can treat them as random draws from a common population distribution. A hierarchical model is a natural way to integrate all of these analyses into a single model.

3.4.2.1 Example of Bayesian Hierarchical Model

Part C: Assuming a 'similar' success rate over several call centers.

Expanding on the example in the last section where the success rate of a new campaign was being studied, let's now assume that 3 other outbound call centers were also measuring the number of successes and the total number of calls.

• Table 5: Data For Call Center Example (Part C)

Call Center (j)	**# of successes (r_j)**	**Total # calls(n_j)**
1	18	30
2	9	20
3	26	44
4	13	25

Y, the observed data, now consists of r_j and n_j for each of the four call centers (j=1,2,3,4). Once again, we will model the number of successes, r_j as a binomial random variate, but this time, we will construct a logistic binomial regression model with 'non-informative' priors as the hyperparameters. The complete model, as built and executed in WinBUGS (Spiegelhalter et al. 1998), is provided in Appendix A.4.

Figure 15 provides a graphical representation of this hierarchical model.

Likelihood: $r_j \sim \text{Binomial}(\theta_j, n_j)$

where $\text{logit}(\theta_j) = \delta$

and $\delta \sim \text{Normal}(\mu, \tau)$

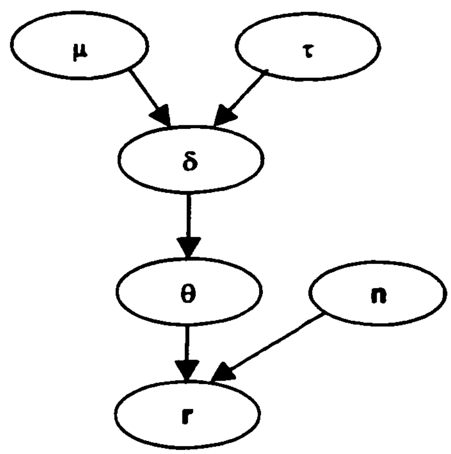
Logit function transforms normal distribution on [0,1]
$\text{logit}(\theta_j)=\delta$ implies
$\log[\theta_j/(1-\theta_j)]=\delta$

Priors: $\mu \sim \text{Normal}(0, 1.0\text{x}10^{-6})$

$\tau \sim \text{Gamma}(1.0\text{x}10^{-3}, 1.0\text{x}10^{-3})$

Reminder: Normal distribution is parameterized in terms of precision. For example, precision = 1.0×10^{-6} is equivalent to a variance of 1.0×10^{6} and a standard deviation of 1000.

• Figure 15: Graphical Model for Example Part C



Hierarchical models such as this are much easier to interpret using simulation methods similar to the ones discussed previously. In the following section, methods of calculating the joint and marginal probabilities of the parameters through Monte Carlo simulation will be discussed.

3.4.3 Markov chain Monte Carlo Methods

Markov chain Monte Carlo (MCMC) has made Bayesian inference accessible to many that would have not used it previously. It has also contributed to the increased acceptance of Bayesian inference since it has almost eliminated the need to over-simplify problems into a framework that could be solved using other, available frameworks. MCMC has its roots in the field of statistical physics, but it has taken nearly 40 years for MCMC to move over to mainstream statistical practice.

MCMC is simply Monte Carlo integration using Markov Chains. Monte Carlo integration was introduced in the previous section and can be briefly explained as repeated sampling from a distribution as an alternative to analytic evaluation of an integral. Monte Carlo simulation is often used to find features of the posterior distribution that may be expressed as expectations of functions of unknown parameter θ. Using Equation 9 as a basis, the expected value of a function of an unknown parameter θ may be written as:

• Equation 12

$$E[f(\theta)\mid Y]=\frac{\int f(\theta)p(\theta)p(Y\mid\theta)d\theta}{\int p(\theta)p(Y\mid\theta)d\theta}$$

Monte Carlo integration evaluates E[*f*(θ)|Y] by drawing k samples from the posterior distribution, p(θ|Y), and then uses the following approximation:

• Equation 13

$$E[f(\theta)\mid Y]\approx\frac{1}{n}\sum_{s=1}^{k}f(\theta^{(s)}) \qquad \textit{where } \theta^{(1)},\theta^{(2)},\ldots,\theta^{(k)}\sim p(\theta\mid Y)$$

When the samples $\{\theta^{(s)}, s = 1,\ldots,k\}$ are independent, the accuracy of the approximation can be improved by increasing sample size, k. With many Bayesian models, the non-standard form of $p(\theta|Y)$ often makes independent sampling from the posterior distribution infeasible. Fortunately, dependent sampling from a Markov chain with $p(\theta|Y)$ as its unique, stationary distribution produces the same successful results. The following section will describe a method to construct a Markov chain such that its stationary distribution is $p(\theta|Y)$.

3.4.4 Gibbs Sampling

The Gibbs sampler is a special case of the Metropolis-Hastings algorithm first introduced by Metropolis et al.(1953) and then later generalized by Hastings(1970). The Gibbs sampler generates a Markov chain by sampling from full conditional distributions. It was given its name by Geman and Geman (1984) and to date, most statistical applications of MCMC have used Gibbs sampling or variations thereof. Gilks, Richardson and Spiegelhalter (1996) provide a full introduction with many worked examples to provide clarity. Gibbs sampling may be summarized as follows:

Let θ be a vector of length q of unknown parameters. $\theta = \{\theta_1, \theta_2,\ldots, \theta_q\}$

1) Choose starting values for each of the unknowns $\theta_1^{(0)}, \theta_2^{(0)}, \ldots, \theta_q^{(0)}$

2) Sample $\theta_1^{(1)}$ from the full conditional distribution $p(\theta_1|\theta_2^{(0)}, \theta_3^{(0)},\ldots, \theta_q^{(0)}, Y)$
   Sample $\theta_2^{(1)}$ from the full conditional distribution $p(\theta_2|\theta_1^{(1)}, \theta_3^{(0)},\ldots, \theta_q^{(0)}, Y)$
   .
   .
   .
   Sample $\theta_q^{(1)}$ from the full conditional distribution $p(\theta_q|\theta_1^{(1)}, \theta_2^{(1)},\ldots, \theta_{q-1}^{(1)}, Y)$

   Repeat step 2 k times

3) After an appropriate *burn-in* period, *k*, a Markov chain is formed with $p(\theta|Y)$ as its stationary distribution.

The length of the burn-in period represents the time it takes for the Markov chain to converge to a stationary distribution. The burn-in period is designed to remove dependence of the simulated chain on its starting values and its length may be anywhere from ten to tens of thousands! Determining convergence of MCMC is still a topic that attracts on-going research (Gelman 1996; Cowles and Carlin 1996) and is discussed further in section 3.4.5.1.

Gibbs samplers have been written in many languages (FORTRAN, SPLUS), but a programming environment developed within the Biostatistics Unit at Cambridge has become a very popular way to implement MCMC using Gibbs Sampling. BUGS (Bayesian inference Using Gibbs Sampling) is a program which provides a syntax for specifying hierarchical models, a command language for running Gibbs Sampling sessions, and reporting features that allow the analyst to evaluate the output. The 'Classic' BUGS program has now been superceded by WinBUGS, as described in a following section.

3.4.5 Current Issues in MCMC

Before demonstrating MCMC using WinBUGS, there are a couple of issues that must be discussed. The question of diagnosing convergence is very important since the results of the simulation will be invalid if statistics are drawn before the Markov chain has converged. Also, if one must decide between two or more alternate models for the same data, how can this be done? These two topics continue to receive a great deal of research attention, so there will be no attempt to propose a definitive solution – rather several options will be discussed and appropriate methods chosen for this application. Kass et al. (1998) present a panel discussion addressing these topics and others.

3.4.5.1 Diagnosing Convergence

Convergence of the Markov chain must be identified in order to determine the length of the *burn-in* period. The easiest and most common method is through simple visual inspection of the chain. While visual inspection of a single chain can accurately determine when a chain has not converged, it is not adequate for confirming convergence. Some chains move through the parameter space very slowly – so much so that they may appear to have converged. This problem occurs frequently since the samples from a Markov chain are serially correlated. Other models, especially high-dimensional models, can be multi-modal, and if the chain is stopped prematurely, the chain may appear to have converged at a mode other than the 'main' mode. Many experts agree (Kass et al. 1998) that it is important to run multiple sampling chains from over-dispersed starting points to ensure that each chain converges to the same value(s). It is also important to ensure that not only have the parameters of interest converged, but also the *nuisance parameters*.

There have been many more formal tools proposed, called *convergence diagnostics*, to estimate the length of burn-in required in order for the chain to 'forget' its starting position. This is still an active area of research with many proposed methods (Geweke 1992; Raftery and Lewis 1992), but an often-used diagnostic is one developed by Gelman and Rubin (1992). This diagnostic, often referred to as $\hat{R}$, is a ratio of estimates of total sequence (pooled) variability and within-sequence variability. The premise is that convergence has been achieved once $\hat{R}$ approaches 1 (total sequence variance $\cong$ within-sequence variance) and the two variances have stabilized. In practice, simulations are run until the values of $\hat{R}$ are all less than 1.2 (Kass et al. 1998).

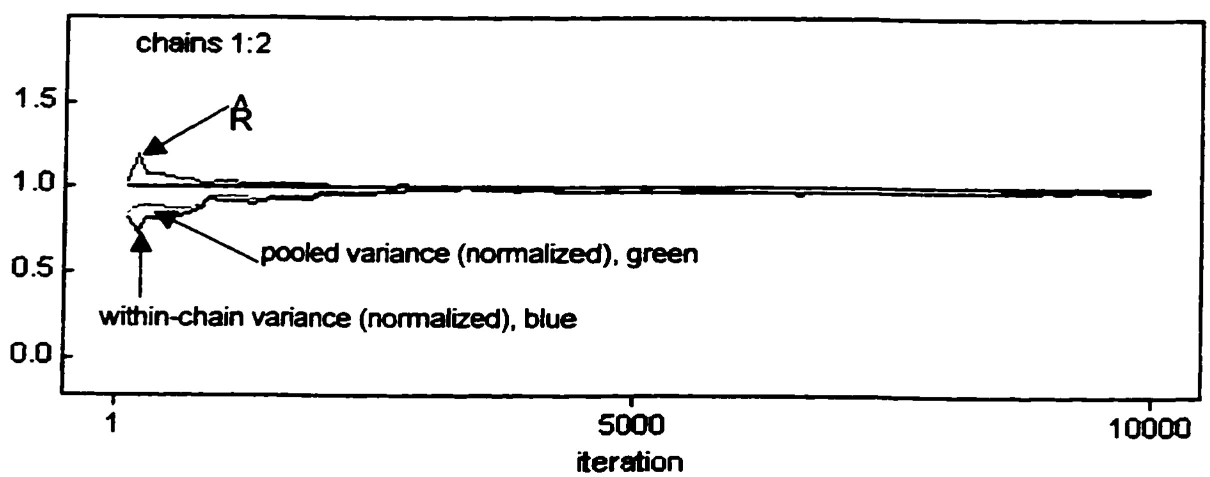
The latest version of WinBUGS calculates the Gelman-Rubin diagnostic (GR-Diag), as modified by Brooks and Gelman (1998). In this case, $\hat{R}$ (see Equation 14, below) is interpreted as the ratio of empirical 80% posterior interval lengths, rather than as a variance ratio.

• Equation 14

$$\hat{R} = \frac{\text{length of 80\% total-sequence interval}}{\text{mean length of the 80\% within-sequence intervals}}$$

As long as multiple chains are run, plots of the convergence statistic are available for any of the monitored parameters. In Figure 16, the GR-Diag is plotted for two chains, each of 10 000 iterations. Since the two variance chains have stabilized and $\hat{R}$ appears to have reached a stable level near 1.0, we can conservatively say that convergence may be assumed after 3000 - 4000 iterations.

• Figure 16: WinBUGS implementation of Gelman-Rubin diagnostic (GR-Diag) for convergence



Others have developed rules to determine *a priori* results about time to convergence for MCMC chains that may be used in conjunction with convergence diagnostics (Roberts and Rosenthal 1998, Rosenthal 1995).

3.4.5.2 Model Selection and Critique

When presented with a Bayesian Model and its output, how can one critique the inferences produced? When two or more competing models are given, how can a choice be made regarding which one is best? Is it reasonable to assume that one model alone is best? These questions are asked by both novices and experts in the field of Bayesian inference, and like the questions raised in the previous section, this research paper will make no attempt to answer these questions.

Classical model selection often analyzes residual errors, that is, the difference between actual and predicted. The models presented will attempt to find the best distribution to predict the individual call handle times. The actual individual call handle times are unavailable (except in the case of the synthetic data), and hence alternatives to the classical methods of model selection are required. Comparisons may be made between predicted and observed aggregated data and it is suggested that this cross-validation be done with separate evaluation data. O'Hagan (1995) advocates the division of data into two parts. The first part may be used as a *training sample* to obtain posterior distributions, and the latter part used for model comparison. Gelman and Meng (1996) use posterior predictive model checking as a method of model critique. In this case, the model is deemed to not fit the data if the actual, observed values for some meaningful *discrepancy variable* are far from the predictive distribution.

Dempster (1974) proposes a method of examining the posterior distribution of the log-likelihood of the observed data. This distribution, often called the *deviance* measure is simple to calculate during a WinBUGS run. The deviance (Equation 15) is re-calculated with each iteration of the converged Markov chain, using the current values of Y and θ.

• Equation 15

$$deviance = -2 \log p(Y \mid \theta)$$

Others (Gilks et al. 1992; Zeger and Karim 1991) used variations of this technique as a method of model comparison. One of the benefits of using *mean deviance* as a method of model critique is not only can it be calculated within the MCMC simulation, but it is also easily understood. The assumption underlying the use of this statistic is the smaller the likelihoods of the observations, the greater the *mean deviance* and hence, the poorer the fit between observed data and model. By choosing the model with the smallest mean deviance, we are attempting to select the model that maximizes the likelihood of observing data Y.

There are many other techniques for model criticism such as Bayes Factors approaches (Berger & Pericchi 1996; O'Hagan 1995, Carlin and Chib 1995), and cross-validatory techniques (Gelfand 1996) which may also be appropriate. Given the simplicity of the *deviance* measure and its immediate availability in WinBUGS, its discriminatory ability should be adequate for the purpose of these call center models.

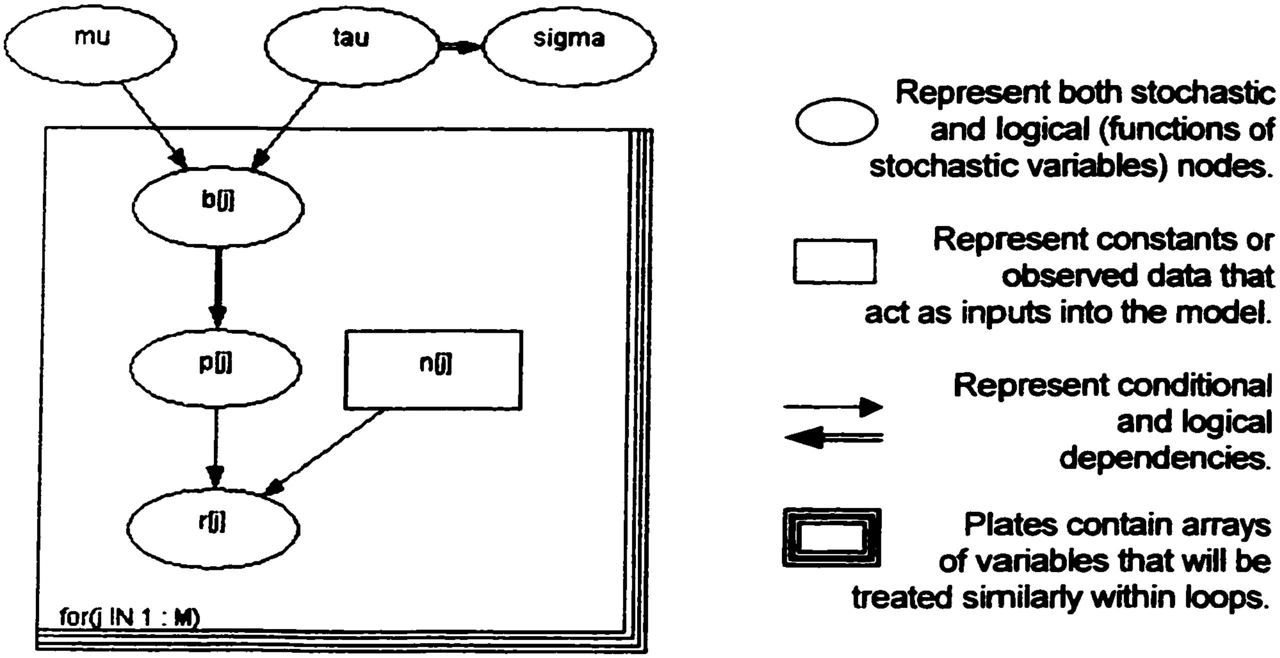
3.4.6 WINBUGS

WinBUGS is an interactive version of the BUGS program. It has similar functionality but it also incorporates a graphical model editor called DOODLEBUGS and on-line viewing of simulations. Documentation and downloadable programs are available from http://www.mrc-bsu.cam.ac.uk/bugs. The models developed for this paper used the recently released, version 1.2 of WinBUGS.

3.4.6.1 Example of Bayesian Inference using WinBUGS

The best way to explain how WinBUGS works is to demonstrate its use through a familiar example. Recall the outbound call centers that want to determine the success rate of a new program at four different locations (see section 3.4.2.1). The first step in WinBUGS is to develop a directed acyclic graphical (DAG) model of the data using the DOODLEBUGS interface (See Figure 17, below):

• Figure 17: DoodleBug for Outbound Call Center Example



Once the DOODLEBUG is complete, WinBUGS can automatically generate the accompanying BUGS code shown in Figure 18 below:

• Figure 18: BUGS code generated from DoodleBUG for outbound call center

```
outbound call center model;
{
  for( j in 1 : M ) {
    r[j] ~ dbin(p[j],n[j])
    logit(p[j]) <- b[j]
    b[j] ~ dnorm(mu,tau)
  }
  tau ~ dgamma(0.001,0.001)
  mu ~ dnorm( 0.0,1.0E-6)
  sigma <- 1 / sqrt(tau)
}
```

Once the code is generated and compiled, the data entered and the initial values set, the next step is to run the Gibbs Sampler and gather statistics concerning the parameters of interest. The latest version of WinBUGS allows the simultaneous analysis of several chains with varying initial values. This feature allows the analyst to monitor several chains on the same chart for convergence assessment and allows the calculation of statistics such as the Gelman-Ruben diagnostic.

For the example given, the data sets and initial values are as follows:

Data r = (18, 9, 26, 13)
n = (30, 20, 44, 25)
M = 4
Inits Chain 1: tau = 10^{-6} mu = 10
Chain 2: tau = 10^{-3} mu = -10

Using two chains, each with a burn-in period of 4000 iterations, statistics were gathered for the remaining 6000 iterations of each chain, producing a sample of 2 x 6000 = 12000. Table 6 lists the summary statistics for all unknown parameters, including the major parameter of interest, p[j] (success rate):

• Table 6: Outbound Call Center Results from WinBUGS

Node	Mean	sd	2.5%	Median	97.5%	Start	sample
b[1]	0.265	0.2432	-0.2072	0.2604	0.7693	4001	12000
b[2]	0.1476	0.2787	-0.4946	0.1724	0.6301	4001	12000
b[3]	0.2662	0.2239	-0.1625	0.262	0.7305	4001	12000
b[4]	0.1954	0.2534	-0.3344	0.2059	0.6631	4001	12000
Mu	0.2183	0.2545	-0.2738	0.2226	0.6842	4001	12000
p[1]	0.5649	0.05856	0.4484	0.5647	0.6834	4001	12000
p[2]	0.5364	0.06787	0.3788	0.543	0.6525	4001	12000
p[3]	0.5653	0.05416	0.4595	0.5651	0.6749	4001	12000
p[4]	0.548	0.06174	0.4172	0.5513	0.66	4001	12000
Tau	198.8	396.6	1.305	53.61	1304.0	4001	12000

The mean success rate for the four call centers ranges from 53.6% to 56.5% and the 95% interval for each is also given. As can be seen from the results, the

Bayesian model produces different mean values for each of the b_j (and hence p_j), but it should be noted that each of the b_j is drawn from the same population: $b_j \sim$ dnorm(μ,σ^2) where $\bar{\mu} = 0.218$ and $\bar{\tau} = 198.8$.

If this model was to be used to predict the success rate at a new call center (p_0), we could use the mean of the distribution from which all b_j belonged, $\bar{\mu} = 0.218$ as an estimate of b_0 and then transform it through the logit function to arrive at p_0.

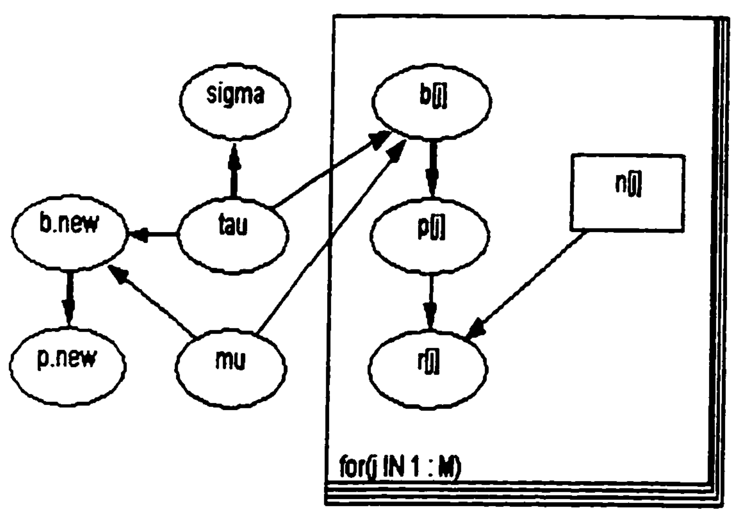
$$b_0 = 0.218 \quad \textit{and} \quad \text{logit}(p_0) = b_0$$

$$\log\frac{p_0}{1-p_0} = b_0$$

$$p_0 = \frac{e^{b_0}}{1+e^{b_0}} = \frac{e^{0.218}}{1+e^{0.218}} = \frac{1.24359}{2.24359} = 0.544$$

An alternate method that is especially useful is demonstrated in Figure 19 below:

• Figure 19: DoodleBUG, BUGS code and new output for outbound call center model II



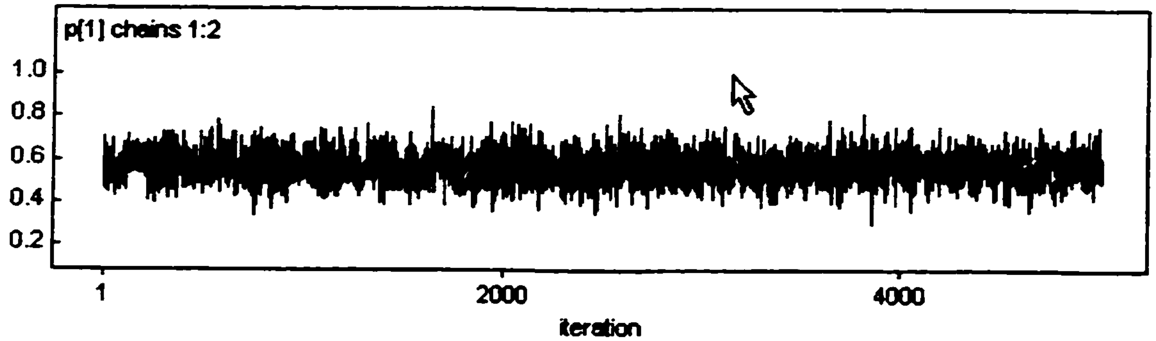

```
outbound call center model II;
{
   for( j in 1 : M ) {
      r[j] ~ dbin(p[j],n[j])
      logit(p[j]) <- b[j]
      b[j] ~ dnorm(mu,tau)
   }
   tau ~ dgamma(0.001,0.001)
   mu ~ dnorm( 0.0,1.0E-6)
   sigma <- 1 / sqrt(tau)
   b.new ~ dnorm (mu,tau)
   logit(p.new) <- b.new
}
```

Node	Mean	sd	2.5%	Median	97.5%	Start	sample
b.new	0.217	0.421	-0.589	0.229	0.985	4001	12000
p.new	0.553	0.090	0.357	0.557	0.728	4001	12000

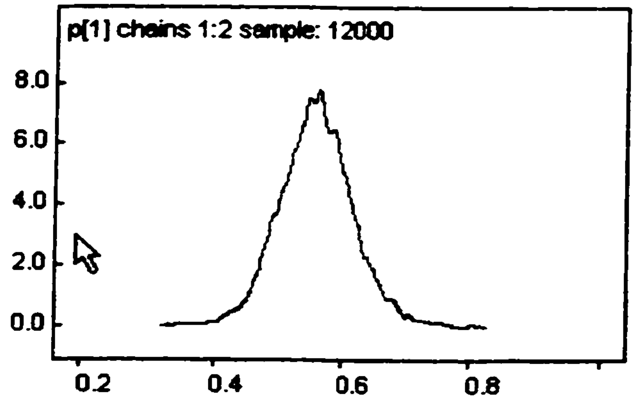
The new parameters called b_{new} and p_{new} are sampled/calculated directly while performing the MCMC simulation. The resulting p.new value indicates that at another call center, the new program would have a prediction interval of [0.357,0.728] with a mean success rate of 55%.

WinBUGS also provides a variety of useful graphical summaries. Analysts often use trace plots of the chains (Figure 20) in order to confirm convergence, kernel density plots such as Figure 21 help to visually represent parameters of interest, and plots of the Gelman-Rubin convergence diagnostic (Figure 22) are used by many before, or instead of, trace plots.

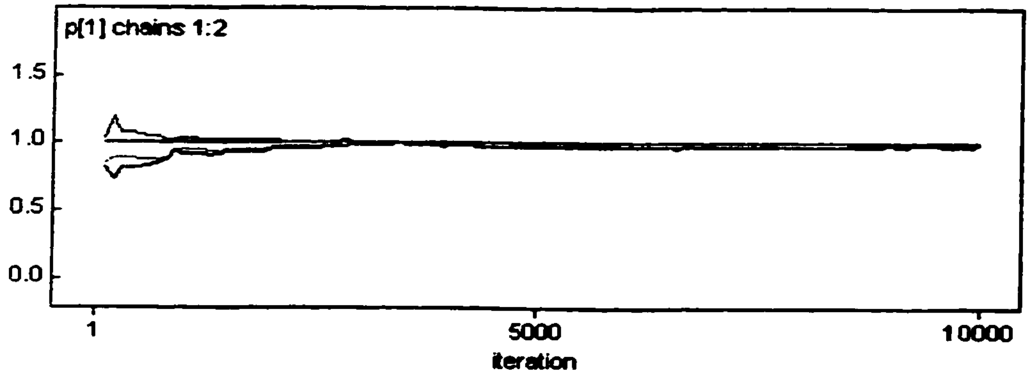
• Figure 20: Trace plot of p[1] for two chains with over-dispersed starting values (5000 iterations)



• Figure 21: Kernel Density Plot for success rate at Call Center One, p[1]



• Figure 22: Plot of Gelman-Rubin convergence diagnostic for p[1]



Sections 3.4.1 through 3.4.6 above provided just a brief description of a few techniques that may be used while analyzing data using Bayesian inference techniques and MCMC. The purpose of the background information was to provide the reader with *just enough* knowledge to understand the models that will be presented in the next section. Those interested in a more thorough analysis are directed to two informative texts: *Bayesian Data Analysis* written by Gelman, Carlin, Stern and Rubin (1995) and *Markov Chain Monte Carlo in Practice* edited by Gilks, Richardson and Spiegelhalter (1996).

3.4.7 Bayesian Models applied to Call Handle Times

Bayesian Inference is the third disaggregation method that will be discussed. The Bayesian model will be used to infer the distribution parameters of individual call handle times as well as provide some insight into model (distribution) selection. This approach was motivated by the unique problem that the individual call handle times are 'missing', although not in the classic statistical sense. Bayesian methods have been quite successful dealing with missing data and led to the contemplation of using these techniques for the problem at hand.

By knowing the average handle time and the number of calls, we are able to calculate the sum of the handle times for each time block. For each of the following models, the observed data Y_i, represents the sum of the n_i calls during time block i. That is, the observed time block average has been transformed into an observed time block sum through the function $Y_i = n_i \overline{X}_i$.

Three models have been created – each reflecting a different assumed theoretical distribution for the underlying individual call handle time. The *exponential model* assumes that each of the arriving calls follows an exponential service time. This is the base model since it reflects current industry standards and has the simplest structure. The *normal model* is presented as an alternative to the exponential model when the coefficient of variation << 1. The final *gamma model* is offered as an approximation to a potential lognormal call handle time distribution. The complete WinBUGS models for each of the three data sets are provided in Appendix A.

3.4.7.1 Exponential Model

If X_i, i=1,…,n are independent identically distributed (iid) exponential random variables each having parameter θ, then $Y = X_1 + X_2 + \ldots + X_n$ is a gamma random variable with parameters (n,θ). This gamma distribution, where n represents an integer, is often referred to as the Erlang distribution. The exponential model is defined with the following distribution:

$$\mathbf{Y_i \sim Gamma(n_i,\theta_i)}$$

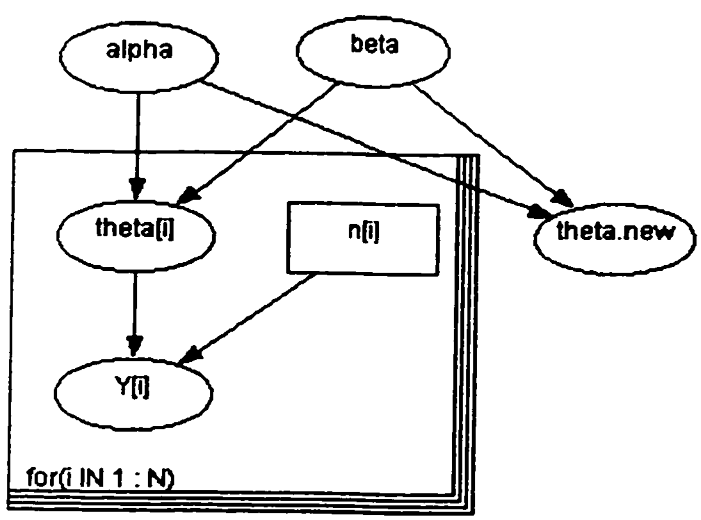
where *n_i is the observed number of calls handled and*
Y_i is the sum of the call handle times in time block i.

The θ_i for each time block are assumed to be from the same distribution, in this case, the gamma distribution with *hyperparameters* alpha and beta. Alpha and beta are given independent, non-informative priors (as discussed in section 3.4.1).

$$\theta_i \sim \mathbf{Gamma(\alpha,\beta)} \quad \textbf{with priors } \alpha,\beta \sim \mathbf{Gamma(0.001,0.001)}$$

The MCMC built in WinBUGS will have p(θ|Y,n) as its stationary distribution. The DoodleBug in Figure 23 represents the hierarchical structure of the model and the accompanying WinBUGS code is also provided.

• Figure 23: DoodleBUG and WinBUGS code for Exponential Model




```
exponential model;
{
 for( i in 1 : N ) {
   Y[i] ~ dgamma(n[i],theta[i])
   theta[i] ~ dgamma(alpha,beta)
   }
 alpha ~ dgamma(0.001,0.001)
 beta ~ dgamma(0.001,0.001)
 theta.new ~ dgamma(alpha,beta)
 mu<-1/theta.new
}
```

Since each of the θ_i are drawn from the same distribution, they will be similar but not identical. The goal of the model is to infer what θ will be for future calls, and the mean of the marginal posterior density of theta.new will be used to predict the parameter of the individual, exponentially distributed, call handle times, θ.

3.4.7.2 Normal Model

If X_i, $i=1,\ldots,n$ are iid normal random variables each having parameters (μ,σ) then $Y = X_1 + X_2 + \ldots + X_n$ is a normal random variable with parameters $(n\mu,\sqrt{n}\sigma)$. The second parameter, standard deviation, must be converted to *precision* in order to be understood by WinBUGS. Precision is the multiplicative inverse of variance, that is, precision = 1/(standard deviation)^2. Hence, if each of the X has a precision of τ then Y has a precision of τ/n_i. The normal model is defined with the following distribution:

$$Y_i \sim (n_i\mu,\tau/n_i)$$

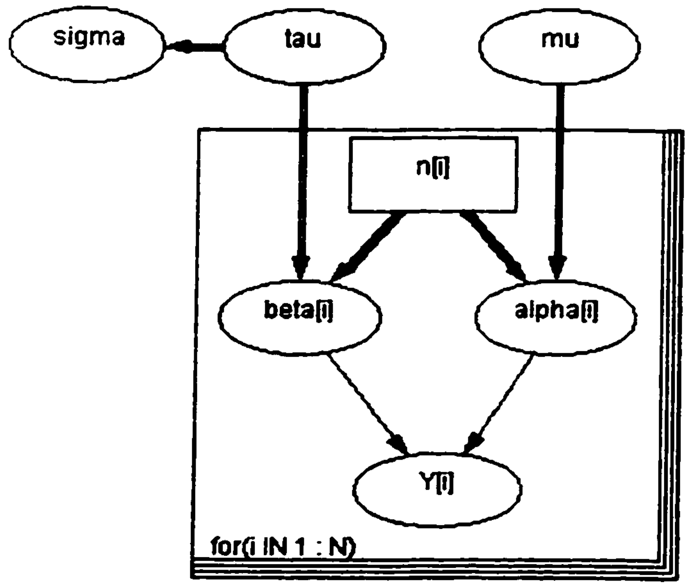
where   *n_i is the observed number of calls handled,*
Y_i is the sum of the call handle times in time block i, and
μ and τ are the mean and precision of the individual calls.

The normal model omitted the second layer of stochastic parameters (hyperparameters) and assumed non-informative 'reference' priors for both mu and tau. This was done merely to demonstrate modeling flexibility rather than one 'correct' method.

Prior distributions: $\mu \sim$ Normal(0,0.000001) and $\tau \sim$ Gamma(0.001,0.001)

The following DoodleBug in Figure 24 represents the hierarchical structure of the model and the generated WinBUGS code is also provided.

• Figure 24: DoodleBUG and WinBUGS code for Normal Model




```
normal model;
{
 for( i in 1 : N ) {
  alpha[i] <- mu * n[i]
  beta[i] <- tau / n[i]
  Y[i] ~ dnorm(alpha[i],beta[i])
 }
 mu ~ dnorm( 0.0,1.0E-6)
 tau ~ dgamma(0.001,0.001)
 sigma <- 1 / sqrt(tau)
}
```

3.4.7.3 Gamma Model

Due to the similar shape and skew, the gamma distribution may be used as a substitute to the lognormal distribution. Since the distribution of the sum of lognormal variates has no closed analytical form, the gamma distribution will be used to approximate the possibility of individual calls following a lognormal distribution.

If X_i, $i=1,\ldots,n$ are independent identically distributed (iid) gamma random variables each having parameters (α, β), then $Y = X_1 + X_2 + \ldots + X_n$ is a gamma random variable with parameters ($n\alpha$, β). This relation is merely a generalization of the one previously used in the exponential model. The gamma model is defined with the following distribution:

$$\mathbf{Y_i \sim Gamma(n_i\alpha,\beta)}$$

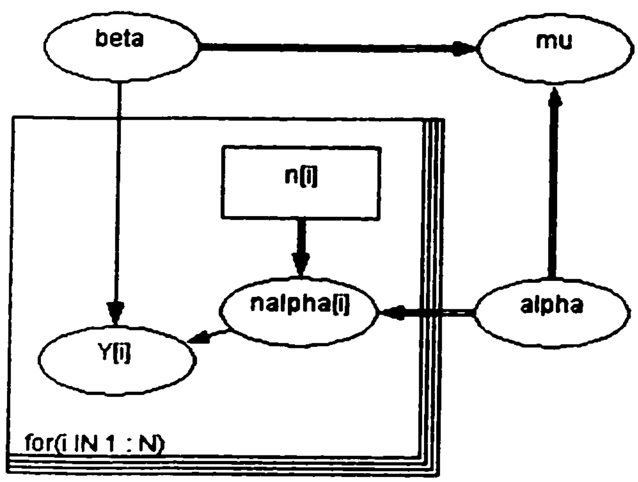
where *n_i is the observed number of calls handled and*
Y_i is the sum of the call handle times in time block i.

The shape(α) and scale(β) parameters for the gamma distribution are given independent, non-informative priors.

α ~ **Gamma (0.001,0.001) and** β ~ **Gamma (0.001,0.001)**

The DoodleBug presented in Figure 25 shows the hierarchical structure of the model and the self-generated WinBUGS code.

• Figure 25: DoodleBUG and WinBUGS code for Gamma Model




```
gamma model;
{
for( i in 1 : N ) {
 nalpha[i] <- n[i] * alpha
 Y[i] ~ dgamma(nalpha[i],beta
 }
  alpha ~ dgamma(0.001,0.001)
  beta ~ dgamma(0.001,0.001)
  mu <- alpha / beta
}
```

The parameter mu (μ =α/β) is calculated at each iteration as a model-checking device. It reflects the current mean of the individual call handle time distribution at each iteration. The implementation of these models is left to the next section.

4.1 Construction of Synthetic Data

An Excel Spreadsheet (see Figure 26) has been created to mimic the data output from an ACD over the course of one eight-hour day. Depending on the distribution being considered, 540 random exponential, normal (censored) or lognormal variables were generated (using Palisades @Risk simulation add-in for Excel) and then aggregated according to the daily fixed call pattern.

- **Data set 1: Exponential (μ = 6.0).** The exponential distribution is often described in terms of the hazard rate, λ, where $\lambda = 1/\mu$. This is the case with WinBUGS. Since the synthetic data consists of 540 random draws from the population, the sample mean and variance may differ slightly from the population:
  - Sample mean: 5.79 Sample standard deviation: 5.89
- **Data set 2: Censored Normal (μ=6, σ=2).** The normal distribution may produce invalid, negative-length call handle times. The frequency of negative-length calls increases as σ gets larger with respect to μ. The ACD simulator recorded a handle time of zero for those instances. With μ=6 and σ=2, this occurred only once for the 540 simulated individual calls.
  - Sample mean: 6.01 Sample standard deviation: 2.04
- **Data set 3: Lognormal(μ=6, σ=4).** The lognormal distribution may be parameterized in numerous ways. In this example, the actual mean and standard deviation of the individual call times are 6 minutes and 4 minutes respectively.
  - Sample mean: 6.04 Sample standard deviation: 4.29

Appendix B gives a full listing of the synthetic data that was used in the examples throughout this thesis.

• Figure 26: ACD Simulator built in Excel

Exponential 6

Raw Data
8.96864
17.2291
9.34082
3.41051
20.7039
1.47265
1.63475
0.25524
10.0074
0.92313
2.16666
4.26984
1.80658
1.24609
4.90044
1.22835
16.1328

Aggregated Data			
time period	#calls	cum #calls	average call length
1	20	0	5.6
2	20	20	5.8
3	30	40	4.6
4	40	70	4.6
5	40	110	6.7
6	40	150	4.9
7	50	190	6.0
8	50	240	4.9
9	40	290	6.8
10	40	330	5.6
11	50	370	5.8
12	40	420	6.0
13	30	460	7.0
14	20	490	4.7
15	20	510	5.6
16	10	530	6.9
	n =	540	

4.2 Application of Disaggregation Models to Synthetic Data

In section 3.2, the 'standard approach' was shown to be inadequate when applied to time-aggregated data. The remaining suggested approaches of using (1) unbiased estimates and (2) Bayesian inference to determine distribution parameters will be applied, analyzed and compared in the following sections.

4.2.1 Unbiased Estimate (UE) Approach

As discussed previously, by using the aggregated data to find the unbiased estimates for mean and variance, we are able to calculate distribution parameters for the individual call time distributions. Using Equation 1 and Equation 7 from section 3.3, the following estimates of population mean and variance were calculated for each of the three sets of synthetic data and displayed in Table 7.

• Table 7: Calculation of UE's for synthetic data and the corresponding Gamma distribution parameters

Call Handle Time Distribution	UE of μ	UE of σ	C.V. σ/μ	Gamma Distribution α $(\mu/\sigma)^2$	Gamma Distribution β (σ^2/μ)
Exponential ($\mu = 6$)	5.79	6.50	1.12	0.8	7.28
Normal ($\mu = 6$, $\sigma = 2$)	6.01	1.68	0.28	12.85	0.47
Lognormal ($\mu = 6$, $\sigma = 4$)	6.03	4.52	0.75	1.79	3.38

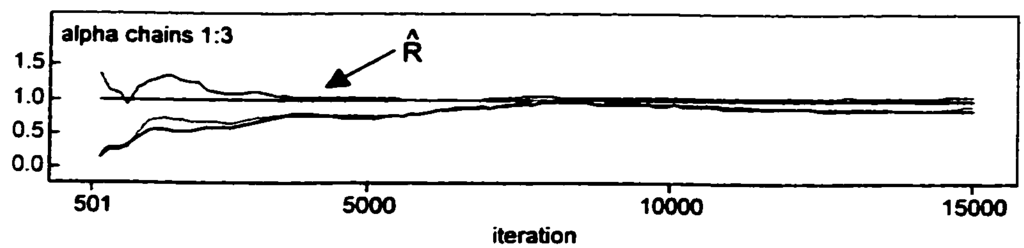
Based on the coefficient of variation, one may be able to make some assumption about the underlying call distribution, but these will always be just informed 'guesses'. For example, the lognormal distribution can take on a wide array of c.v.'s, so it would often be impossible to discern it from the normal, the exponential or many other continuous distributions. In the absence of any other data supporting one distribution over another, it would be helpful to have a 'standard approach'.

Since the gamma distribution is very flexible, it can be used to approximate a variety of distributions. For example, if the data suggests a coefficient of variation of 1, then the gamma distribution would have 1 as its shape (α) parameter, and behave as an exponential distribution. By assuming a gamma distribution, we are, in a sense, allowing the data to choose the shape of the distribution. The Gamma parameter estimates can be calculated directly from the estimated population mean and standard deviation. These parameters, based on the unbiased estimated values of μ and σ, are also provided above in Table 7.

4.2.2 Bayesian Approach

For each of the three models presented in the previous sections, the three synthetic data sets were used as input into the models: exponential, normal and lognormal (individual) call handle times. For each model, three to five overdispersed sampling chains were used, each ranging from 15 000 to 30 000 iterations. Gelman-Rubin Diagnostics were plotted and convergence was assumed after the ratio of pooled variance to within-chain variance ($\hat{R}$) fell below 1.2. (See Figure 27 below)

• Figure 27: Plot of GR-Diag from WinBUGS – convergence assumed after 5000 iterations



A variety of parameters were sampled from the MCMC chain once convergence was achieved. For a complete listing of outputs, please refer to the WinBUGS documents in Appendix A. One of the monitored statistics was the negative log-likelihood of the observed data, called *deviance*. For each model/data combination, the predictive model parameters and the deviance are given in Table 8.

Once the model had been fit to the data, the next step was to determine how the information from the model could be used to predict future individual call lengths. Standard call center simulation packages are not (yet) set up to accept hyperparameters and density functions as distribution parameters – these packages require point estimates of the parameters. To this end, the means of the posterior parameter distributions were used as point estimates of the parameter(s) of each predictive model. For example, the predictive exponential distribution model used the mean of the posterior distribution of theta.new as its single parameter while the actual mean of the alpha and beta parameters were used to parameterize the Gamma distributions.

• Table 8: Output from the three MCMC models for each of three synthetic data sets.

Model↓ Data→	Exponential(6)	Normal(6,2)	Lognormal(6,4)
Exponential	E(5.71) Mean deviance {155.3}	[illegible]	[illegible]
Normal	[illegible]	N(6.01,1.77) Mean deviance {118.0}	[illegible]
Gamma	[illegible]	G(12.37,0.49) Mean deviance {117.6}	G(1.72,3.50) Mean deviance {149.2}

The *mean deviance* statistic was used to determine the dominated models, shaded grey and the 'best' (non-dominated) models, highlighted in white. For the normal models, the deviance statistic was <u>not</u> used exclusively since those models whose tail extended far into the negative call lengths would have an understated deviance. This effect is caused by the absence of observed data in the low-likelihood region of the far-left tail. The suggested normal model for the lognormal data had a mean of 6.04 and a standard deviation of 4.75. Over ten percent of all calls produced by this model would have a negative call length! For the normal models, a visual check of the distribution should be sufficient to ensure that the negative tail area is not significant before choosing the model.

4.2.3 Comments on the Application of Methods

In summary, both the UE method and the Bayesian method were able to disaggregate the data. While the UE method is considerably easier to implement, it still requires the analyst to first choose a theoretical distribution to which to fit the data. It is proposed that the Gamma distribution is suitably flexible for this task. The Bayesian approach is also not without its own subjectiveness. By fitting the data to all three models, the analyst is then faced with model selection. It has been proposed that the *deviance* statistic, easily calculated during the MCMC simulation, be used to help choose the 'best' model, but this is a rudimentary measure, and there are a variety of other sampling-based techniques that may be applied (Gelfand 1996).

4.3 Model Testing

For each of the three data sets, at least 2 distributions have been proposed as best representing the individual call handle times. A summary of the models to be tested is given in Table 9 below.

• Table 9: Disaggregation models chosen to be tested

Synthetic Data Set	Bayesian (MCMC) Method	Unbiased Estimator (UE) Method
Exponential (μ=6)		
	Exponential (μ=5.71)	Gamma(α=0.8, β=7.28)
Normal (μ=6, σ=2)		
	Gamma (α=12.37, β=0.49)	Gamma (α=12.85, β=0.47)
	Normal (μ=6.01, σ=1.77)	
Lognormal (μ=6, σ=4)		
	Gamma (α=1.72, β=3.50)	Gamma (α=1.79, β=3.38)

In order to determine the effectiveness of these models, an experimental design was required that considered the ultimate purpose of the distribution of individual calls: call center simulation.

The call center manager uses the simulation to get an accurate estimate of critical performance measures. Two important call center performance measures that are used industry-wide are service level (%) and abandonment (%). Call centers are often remunerated based on their ability to maintain acceptable levels of these measures. To test the disaggregation models, it is proposed that the model has been successful if its use during call center simulations produce very similar performance estimates to the simulations using the known actual call handle times.

4.3.1 A Simple Call Center Simulation

Using Call$im, a call center simulation package that is built on Systems Modeling's Arena simulation software, a simple call center model was created. (see Appendix C for details) The test call center employs 10 TSRs that work from 9 am to 5 pm with no breaks. This obviously non-unionized call center receives calls randomly throughout the 8-hour day according to the call pattern in Figure 3. The daily call volume is 540 calls and the call center wants to maintain an average service of 80% (In this case, 80% of all calls answered within.

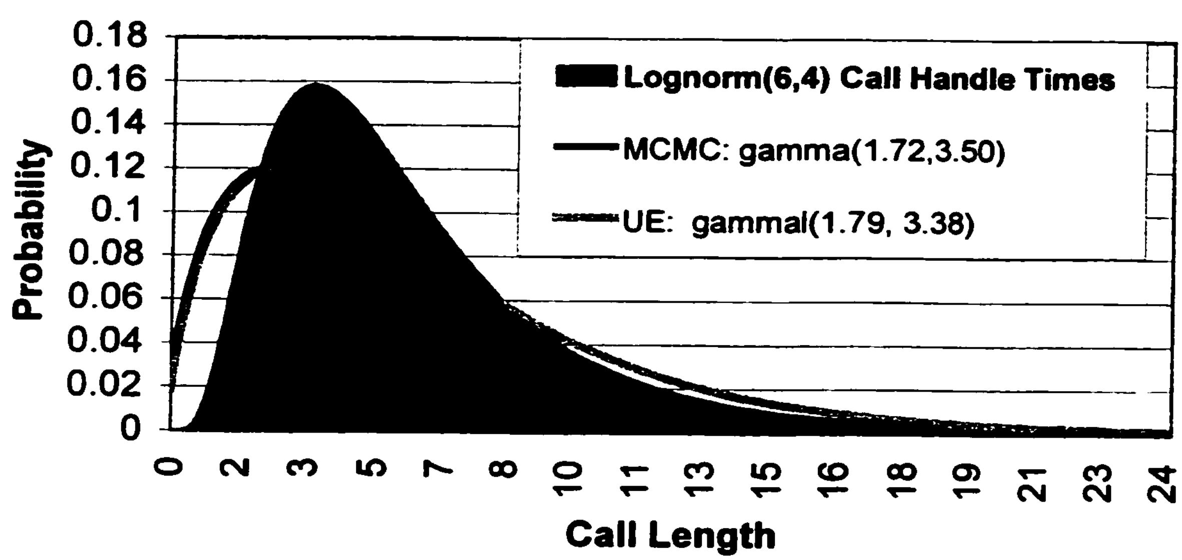
Those calling the call center will abandon after waiting a certain amount of time, a random normal variate (truncated at zero) with a mean of 3 minutes and a standard deviation of 1 minute. There is no after-call work associated with the calls arriving at this call center and hence, call handle time is call talk time. For each simulation, the only variable that changes is the distribution of call handle times. 400 iterations were used in each simulation in order to attain a half-width of approximately 0.5%. The results of the simulations will be provided in the following section.

Chapter 5 Results and Discussion

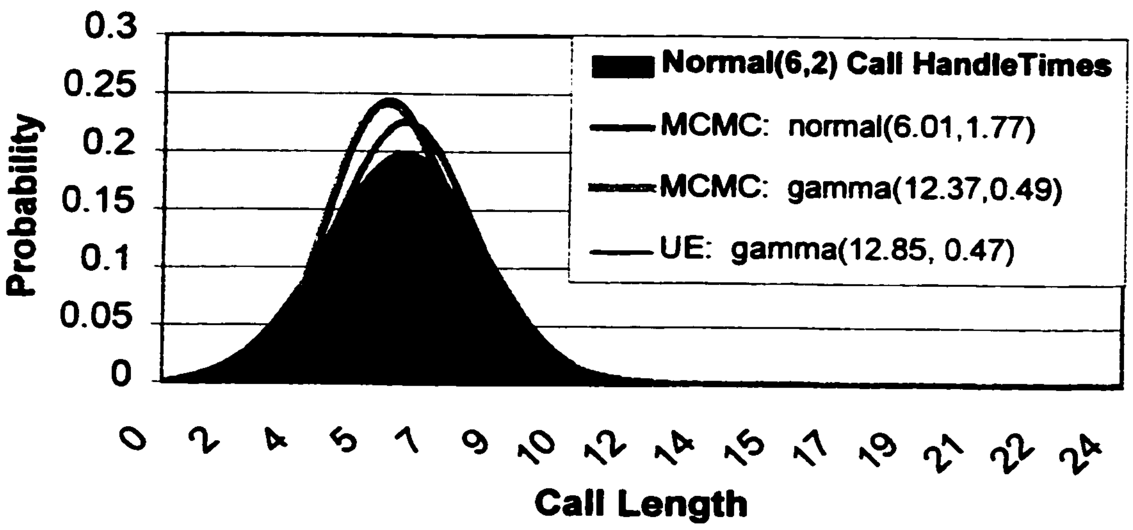
5.1 A Visual Comparison

While certainly not scientific, the first natural tendency is to use a simple visual inspection of the known individual call handle time distributions and the distributions proposed by the models. To satisfy that curiosity, the following graphs are provided to compare the models proposed for the lognormal(6,4) call handle times (Figure 28), the censored normal(6,2) call handle times (Figure 29) and the exponential(6) call handle times (Figure 30).

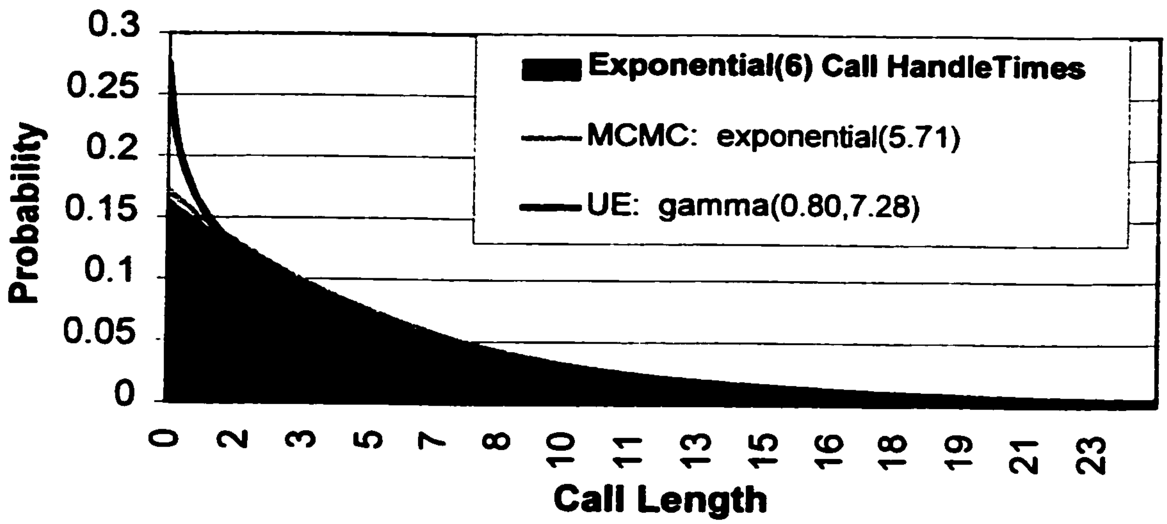
• Figure 28: Proposed Models for lognormal call handle times



• Figure 29: Proposed Models for normal call handle times



• Figure 30: Proposed Models for exponential call handle times



The distributions proposed by both the UE method and the Bayesian (MCMC) method are displayed together with the actual, individual call handle time distribution.

5.2 Call Center Simulation Results

Since the goal of determining the distribution of individual call lengths is use it as input into a call center simulation, one method of determining the effectiveness of the method is to compare results to the known actual.

Using the call center model described in section 4.3.1, the summary results in Table 10 were calculated based on 400 iterations and the various handle time distributions listed. For each of the three sets of synthetic data, simulations were produced using actual and modeled call handle times (MCMC method and UE method) as well as using the current industry practice of modeling call handle times using the exponential distribution.

• Table 10: Comparison of simulation results for actual and modeled call handle times

Handle Time Distribution		Service Level		Abandonment Rate	
		Mean	**% Error**	**Mean**	**% Error**
Exponential (μ=6)	*actual*	80.1%		5.3%	
Exponential (μ=5.79)	*industry*	82.2%	2.6%	4.2%	-20%
Exponential (μ=5.71)	*mcmc*	82.5%	3.0%	4.1%	-22%
Gamma(α=0.8, β=7.28)	*ue*	81.8%	2.2%	4.7%	-11%
Normal (μ=6, σ=2)	*actual*	77.6%		4.0%	
Exponential (μ=6.01)	*industry*	79.7%	2.7%	5.2%	29%
Normal (μ=6.01, σ=1.77)	*mcmc*	77.5%	-0.2%	4.0%	1%
Gamma (α=12.37, β=0.49)	*mcmc*	81.0%	4.4%	5.9%	47%
Gamma (α=12.85, β=0.47)	*ue*	81.4%	4.9%	5.9%	46%
Lognormal (μ=6, σ=4)	*actual*	78.4%		4.4%	
Exponential (μ=6.04)	*industry*	79.4%	1.2%	5.2%	19%
Gamma (α=1.72, β=3.50)	*mcmc*	77.9%	-0.7%	4.3%	-1%
Gamma (α=1.79, β=3.38)	*ue*	77.6%	-1.1%	4.4%	1%

The first observation is that both methods appear to have been successful in arriving at plausible 'disaggregated' individual handle time distributions that perform well in simulations against the known actuals. A secondary and equally important observation is that if individual call handle times were in fact Normal(6,2) or Lognormal(6,4), then the industry standard of modeling the call handle times with an exponential distribution would significantly overestimate service levels. This could easily lead to understaffing and potential sanctions and/or poor customer service levels.

5.3 Implications of Research and Future Opportunities

The initial success of the UE method is very encouraging and can, with little effort, be applied to call center input analysis. This method uses the aggregated data to calculate the unbiased estimates of population mean and variance, and subsequently derives the parameter estimates for the Gamma distribution using estimated values of μ and σ. In the above examples, only 16 data points were used -- representing a single day's worth of aggregated data. It can be expected that analysts would have much more data available to them and hence, better estimates of μ and σ.

Analysts that still hold on to the belief that service times must be exponential may use this method knowing that if the data suggests a c.v. = 1.0, then an exponential call handle time distribution will result. If the c.v. $\neq$ 1.0 then it would still be prudent for the analyst to use both handle times as a form of sensitivity analysis.

The effectiveness of the MCMC disaggregation technique is also impressive, but the learning curve and computational effort required to implement this technique are prohibitive. This method should not be discarded though, since its strength may lie in the modeling of more complex call handle time distributions rather than the simple examples chosen for this research. For example, if it is shown that the call handle time is a mixture (of unknown proportions) of two or more distinct distributions as suggested by Bolotin (1994) then the Bayesian MCMC disaggregation technique may be the only viable approach. Also, if the assumption of a single call handle time distribution across all time blocks is relaxed, a Bayesian approach would be well-suited.

Of great interest to practitioners, a better understanding of call handle time distributions is essential. A time study of this distribution, across many industry sectors, is required and would be a logical next step in this line of research.

Bibliography

1. Abeysinghe, T. and Lee, C. 1998. Best linear unbiased disaggregation of annual GDP to quarterly figures: The case of Malaysia. *Journal of Forecasting*, **17**, 527-537

2. Ahn, J-H. and Ezawa, K.J. 1997. Decision support for real time telemarketing operations through Bayesian network learning. *Decision Support Systems*, **21**, 17-27.

3. Banks, J. and Carson, J.S. 1984. *Discrete event system simulation*. Prentice Hall, New Jersey.

4. Berger, J.O. and Pericchi, L.R. 1996. The intrinsic Bayes factor for model selection and prediction. *Journal of the American Statistical Association*, **91**, 109-122.

5. Best, N.G., Cowles, M.K. and Vines, S.K. 1995. *CODA: Convergence Diagnosis and Output Analysis Software for Gibbs sampling output, Version 0.3*. MRC Biostatistics Unit, Cambridge

6. Bolotin, V.A. 1994. Telephone circuit holding time distributions. In *Proceedings of the 14th International Teletraffic Congress* (ed. J. Labetoulle and J.W. Roberts)

7. Carlin, B.P. and Chib, S. 1995. Bayesian model choice via Markov chain Monte Carlo methods. *J. R. Statist. Soc. B*, **57**, 473-484.

8. Cheng, R.C.H. 1994. Selecting input models. In *Proceedings of the 1994 Winter Simulation Conference* (ed. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila), IEEE Piscataway, New Jersey, 184-191.

9. Cheng, R.C.H. and Holland, W. 1997, Sensitivity of computer simulation experiments to errors in input data. *J. Statist. Comput. Simul.*, **57**, 219-241.

10. Chick, S.E. 1997a. Bayesian analysis for simulation input and output. In *Proceedings of the 1997 Winter Simulation Conference* (ed. S. Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson), IEEE Piscataway, New Jersey, 253-260.

11. Chick, S.E. 1997b. Selecting the best system: A decision-theoretic approach. In *Proceedings of the 1997 Winter Simulation Conference* (ed. S. Andradottir, K.J. Healy, D.H. Withers, and B.L. Nelson), IEEE Piscataway, New Jersey.

12. Chick, S.E. 1998. The effect of input distribution selection on the output of stochastic simulations: A Bayesian perspective. To appear.

13. Chow, G.C. and Lin, A. 1976. Best linear unbiased estimation of missing observations in an economic time series, *Journal of the American Statistical Association*, **61**, 719-721

14. Cleveland, B. and Mayben, J. 1997. *Call Center Management on Fast Forward: Succeeding In Today's Dynamic Inbound Environment*. Call Center Press, Annapolis, Maryland.

15. Draper, D. 1995. Assessment and propagation of model uncertainty. *J.R. Statist. Soc. B.*, **57**, 45-97.

16. Evans, M., Hastings, N., and Peacock, B. 1993. *Statistical Distributions, 2nd Edition,* John Wiley and Sons, New York

17. Evans, M. and Swartz, T. 1995. Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems. *Statistical Science*, **10**, 254-272.

18. Gelfand, A.E. 1996 Model determination using sampling-based methods. In *Markov chain Monte Carlo in Practice.* (ed. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter), Chapman & Hall, London.

19. Gelman, A. 1996. Inference and monitoring convergence. In *Markov chain Monte Carlo in Practice.* (ed. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter), Chapman & Hall, London.

20. Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. 1995. *Bayesian Data Analysis*, Chapman & Hall, London.

21. Gelman, A. and Meng, X-L. 1996. Model checking and model improvement. In *Markov chain Monte Carlo in Practice.* (ed. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter), Chapman & Hall, London.

22. Gilks, W.R., Richardson, S. and Spiegelhalter, D.J. 1996. Introducing Markov chain Monte Carlo. In *Markov chain Monte Carlo in Practice.* (ed. W.R. Gilks, S. Richardson, and D.J. Spiegelhalter), Chapman & Hall, London.

23. Grossman, T.A., Oh, S.L., Rohleder, T.R. and Samuelson, D. 1999a. Call Centers. In *The Encyclopedia of Operations Research and Management Science, 2nd Edition.* (ed. S.I. Gass and C.M. Harris).

24. Grossman, T.A., Oh, S.L., Rohleder, T.R. 1999b. Operations Management for Incoming Call Centers: Industry Practice, Literature Survey and Research Needs. Working paper, Faculty of Management, University of Calgary.

25. Guttman, I., Wilks, S.S., and Hunter, J.S. 1982. *Introductory engineering statistics, 3rd Edition.* John Wiley & Sons, New York.

26. Hsiao, C. 1979. Linear regression using both temporally aggregated and temporally disaggregated data. *Journal of Econometrics*, **10**, 243-252.

27. Jonsson, H. and Silver, E.A. 1987. Modification of Demand Distributions on the Basis of Aggregate Information. *IEE Transactions*, **19**, 379-384

28. Kaplan, A.J. 1988. Bayesian approach to inventory control of new parts. *IIE Transactions,* **20**, 151-156.

29. Kass, R.E. and Wasserman, L. 1996. The selection of prior distributions by formal rules. *Journal of the American Statistical Association,* **91**, 1343-1370

30. Kass, R.E., Carlin, B.P., Gelman, A. and Neal, R.M. 1998. Markov chain Monte Carlo in Practice: A roundtable discussion. *The American Statistician*, **52**, 93-100.

31. MacPherson, Jr. G. F. 1988. *What Senior Managers Need to Know About Incoming Call Centers.* SLN Inc., Annapolis, Maryland.

32. McGrath, M.F., Gross, D. and Singpurwalla, N. 1987. A subjective Bayesian approach to the theory of queues I – modeling. *Queueing Systems*, **1**, 317-333.

33. Mehrotra, V. 1997. Ringing Up Big Business. *OR/MS Today,* **24(4),** 18-24

34. Methrotra V., Profozich, D., and Bapat, V. 1997. The Best Way to Design Your Call Center. *Telemarketing and Call Center Solutions,* **16(5),** 28-29,128-129.

35. Neal, R.M. 1993. Probabilistic inference using Markov chain Monte Carlo methods, technical report CRG-TR93-1, University of Toronto, Dept. of Statistics.

36. O'Hagan, A. 1995. Fractional Bayes factors for model comparison. *J. R. Statist. Soc. B*, **57,** 99-138.

37. Palm, F.C. and Nijman, T.E. 1982. Linear regression using both temporally aggregated and temporally disaggregated data. *Journal of Econometrics*, **19**, 333-343.

38. Rahko, K. 1991. Measurements for control and modeling of teletraffic. In *Proceedings of the 13th International Teletraffic Congress.* Copenhagen.

39. Roberts, G.O. and Rosenthal, J.S. 1998. Markov-chain Monte Carlo: Some practical implications of theoretical results. *The Canadian Journal of Statistics*, **26**, 5-31

40. Rosenthal, J.S. 1995. Rates of convergence for Gibbs sampling for variance component models. *Ann. Statist.*, **23**, 740-761

41. Silver, E.A. and Fiechter, C-N. 1995. Preventive maintenance with limited historical data. *European Journal of Operational Research*, **82**, 125-144.

42. Spiegelhalter, D.J., Thomas, A., Best, N.G., and Gilks, W.R. 1996. *BUGS: Bayesian inference Using Gibbs Sampling, Version 0.5.* MRC Biostatistics Unit, Cambridge, UK.

43. Spiegelhalter, D.J., Thomas, A., and Best, N.G. 1998. *WinBUGS User Manual, Version 1.1.1.* MRC Biostatistics Unit, Cambridge, UK.

44. Spiegelhalter, D. 1999. Bayesian Statistical Analysis Tutorial Notes. Presented at *7th International Workshop of Artificial Intelligence and Statistics.* January 3, 1999. MRC Biostatistics Unit, Cambridge, UK.

45. Takemori, E., Usui, Y. and Matsuda, J. 1985. Field data analysis for traffic engineering. In *Proceedings of the 11th International Teletraffic Congress.* Tokyo.

Appendices

Appendix A: WinBUGS Models

A.1 WinBUGS Model based on Exponential Call Handle Times

A.2 WinBUGS Model based on Normal Call Handle Times

A.3 WinBUGS Model based on Gamma Call Handle Times

A.4 WinBUGS Model of Outbound Call Center Example

Appendix B: Summary of Synthetic Data

B.1 Summary of Simulated ACD Data

B.2 Exponential Data Detail

B.3 Normal Data Detail

B.4 Lognormal Data Detail

Appendix C: Simulations using distributions/parameters from Bayesian Models

C.1 Call Center Simulation with underlying Normal Call Handle Times

C.2 Call Center Simulation with underlying Exponential Call Handle Times

C.3 Call Center Simulation with underlying Lognormal Call Handle Times

Appendix A.1

WinBUGS Model based on Exponential Call Handle Times

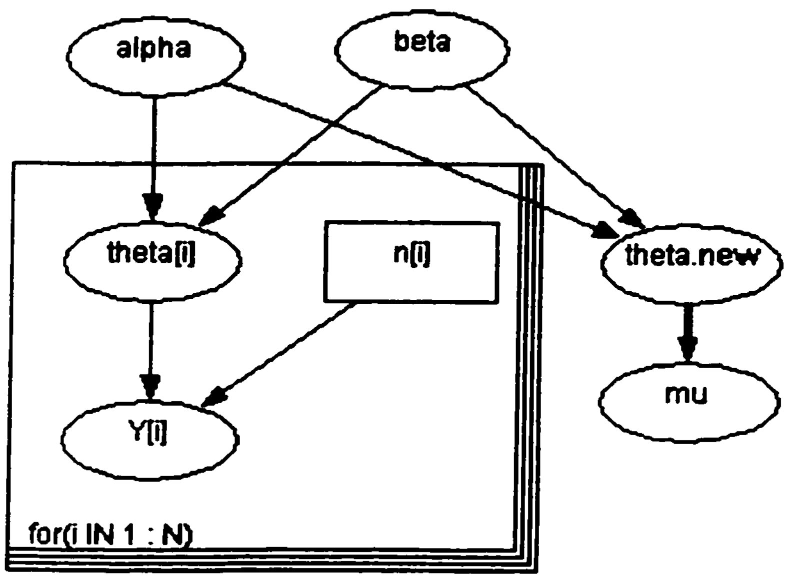
The model has been defined with the following distributions:

Y[i] ~ G(n[i], theta[i])

theta[i] ~ G(alpha, beta)

alpha and beta are given independent, non-informative priors.

Graphical Call Distribution Model



Bugs Language for Call Distribution Model

```
for( i in 1 : N ) {
    Y[i] ~ dgamma(n[i],theta[i])
    theta[i] ~ dgamma(alpha,beta)
  }
  alpha ~ dgamma(0.001,0.001)
  beta ~ dgamma(0.001,0.001)
  theta.new ~ dgamma(alpha,beta)
  mu<-1/theta.new
}
```

Data(Dataset1)

list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(146,95,210,265,186,191,299,364,230,273,235,172,184,80,127,71))

Data(Dataset2)

list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(123,119,166,237,230,240,296,293,235,255,305,261,184,133,111,58))

Data (Dataset3)

list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(145,96,208,268,293,217,267,290,216,250,327,234,155,99,138,57))

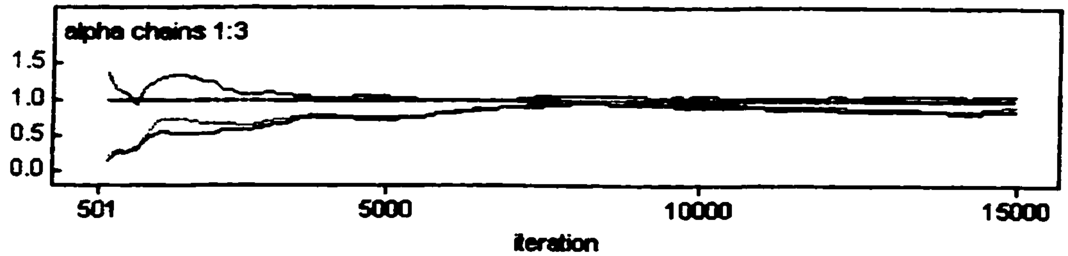
Inits(Chain1) list(alpha=1.0,beta=1.0)

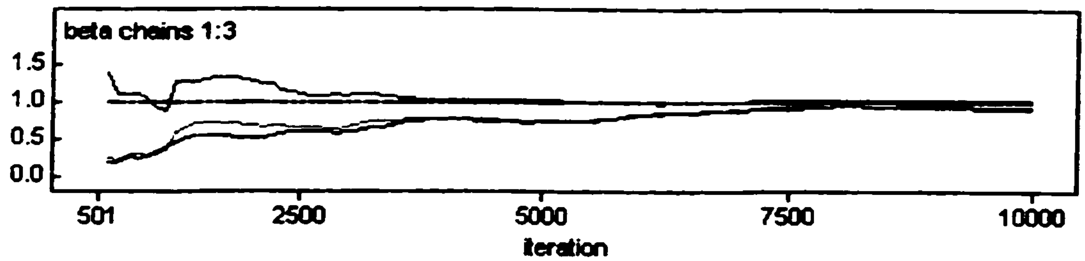
Inits(Chain2) list(alpha=0.10,beta=10)

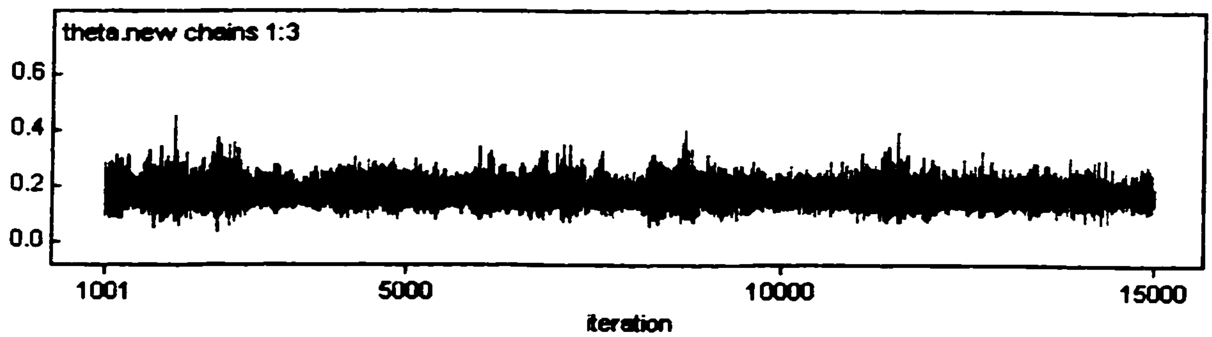
Inits(Chain3) list(alpha=10,beta=0.10)

Results(Dataset 1)

First, Gelman Rubin Diagnostic statistics to determine convergence. A burn-in period of 10000 iterations was used.







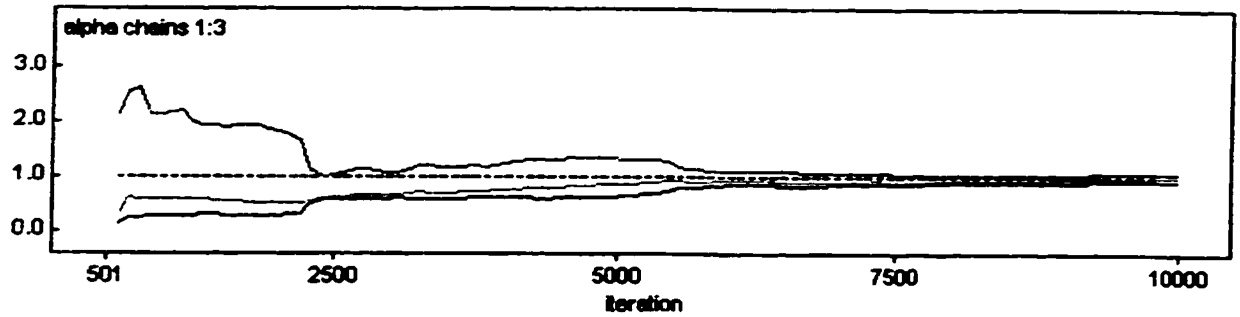
A 10000 update burn-in followed by a further 15000 updates (over 3 chains) gave the following parameter estimates:

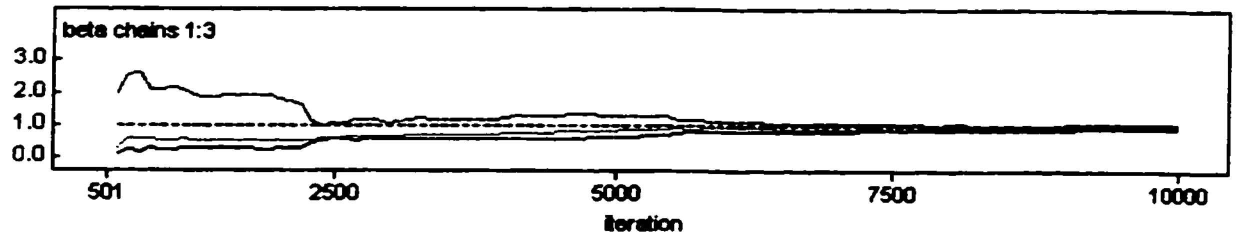
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	99.87	95.6	8.321	17.06	65.3	388.7	10001	45000
beta	574.5	553.9	48.23	96.06	372.7	2237.0	10001	45000
deviance	155.3	4.277	0.1407	147.5	155.1	164.1	10001	45000
mu	5.842	0.9084	0.008719	4.296	5.755	7.9	10001	45000
theta.new	0.1751	0.02614	2.507E-4	0.1266	0.1738	0.2328	10001	45000

The underlying call length distribution in this example was Exponential(mu=6)
The raw (unaggregated) data had an actual mean of 5.79 and a s.d. of 5.89

Results (Dataset 2)

First, Gelman-Rubin diagnostics are calculated and plotted to determine convergence. A burn-in period of 10000 iterations was used.

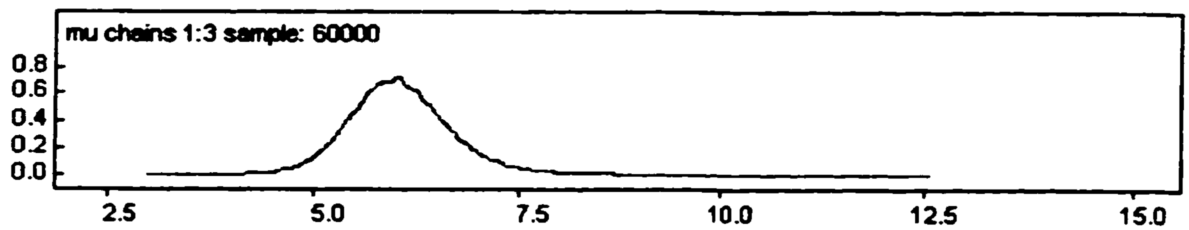




A 10000 update burn-in followed by a further 20000 updates (over each of 3 chains) gave the following parameter estimates:

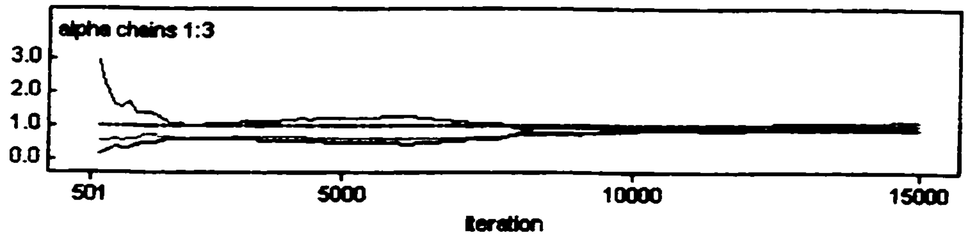
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	174.5	112.1	8.735	38.1	147.0	482.1	10001	60000
beta	1047.0	675.7	52.73	226.5	877.8	2871.0	10001	60000
deviance	146.6	2.443	0.1014	143.4	146.1	152.8	10001	60000
mu	6.06	0.6444	0.00637	4.924	6.016	7.473	10001	60000
theta.new	0.1668	0.01742	1.6E-4	0.1338	0.1662	0.2031	10001	60000

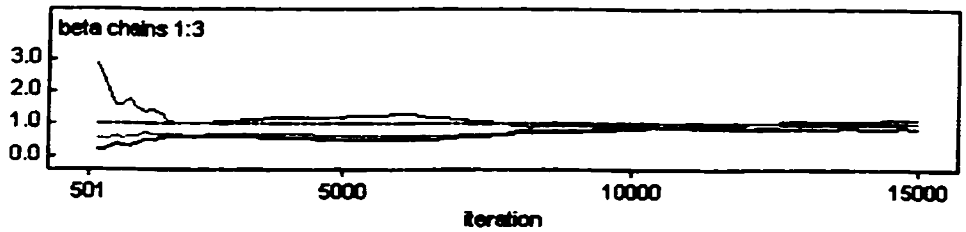
The underlying call length distribution (population) in this example was Normal(mu=6, sigma = 2). The raw (unaggregated) data had an actual mean of 6.01 and s.d. of 2.04.

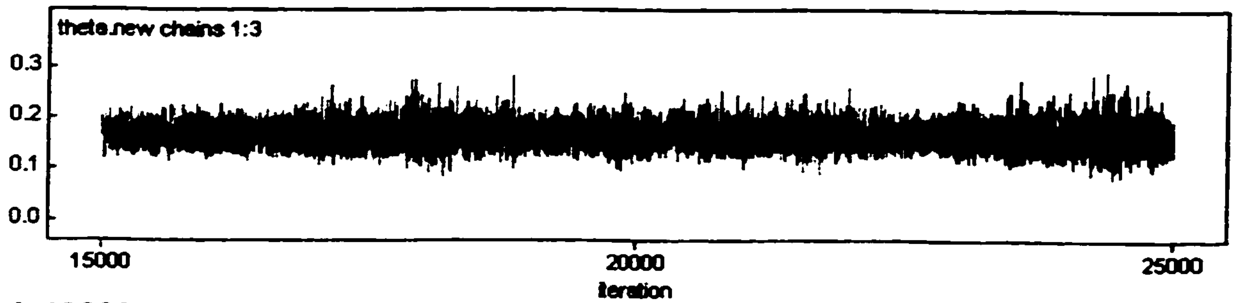


Results (Dataset3)

First, Gelman-Rubin Diagnostic statistics are calculated and plotted to determine convergence.







A 10000 update burn in followed by a further 20000 updates (over each of 3 chains) gave the following parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	149.5	81.66	6.327	35.52	136.2	344.1	10001	60000
beta	899.2	491.5	38.1	212.1	817.7	2054.0	10001	60000
deviance	151.1	2.803	0.02957	146.5	150.7	157.5	10001	60000
mu	6.074	0.6755	0.005623	4.877	6.028	7.541	10001	60000
theta.new	0.1666	0.01824	1.506E-4	0.1326	0.1659	0.205	10001	60000

The underlying call length distribution in this example was Lognormal(mu=6, sigma=4). The individual call handle times had an actual mean of 6.04 and a s.d. of 4.29

Appendix A.2

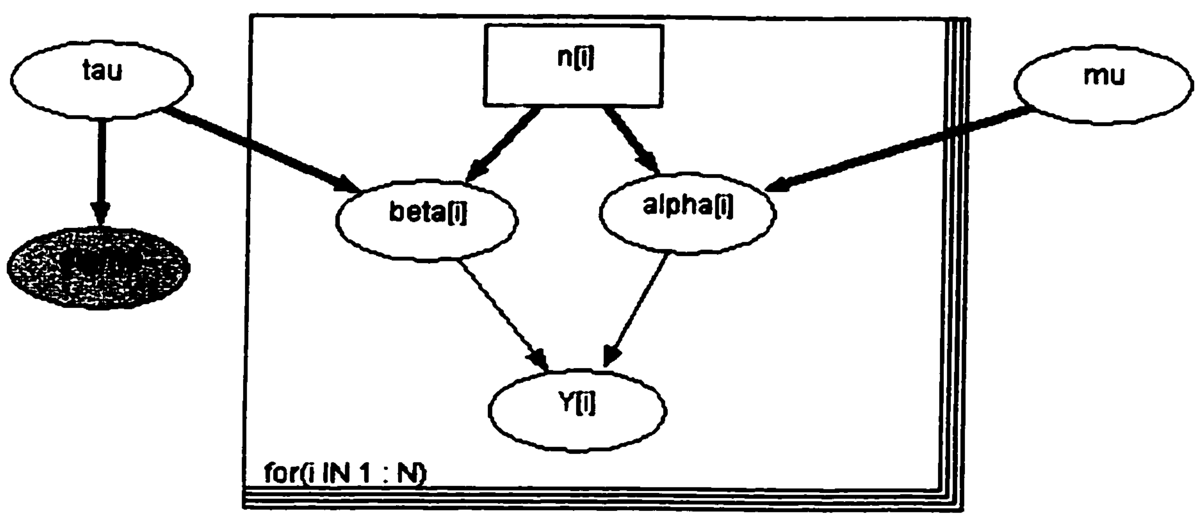
WinBUGS Model based on Normal Call Handle Times

The model has been defined with the following distributions:

$$Y_i \sim \text{Normal}(n_i\mu,\ \tau/n_i)$$

mu and tau are given independent, non-informative priors.

Graphical Call Distribution Model



Bugs Language for Call Distribution Model

```
model;
{
  for( i in 1 : N ) {

    alpha[i] <- mu * n[i]
    beta[i] <- tau / n[i]
    Y[i] ~ dnorm(alpha[i],beta[i])
  }
  mu ~ dnorm( 0.0,1.0E-6)
  tau ~ dgamma(0.001,0.001)
  sigma <- 1 / sqrt(tau)
}
```

Data(Dataset1)

```
list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(146,95,210,265,186,191,299,364,230,273,235,172,184,80,127,71))
```

Data(Dataset2)

```
list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(123,119,166,237,230,240,296,293,235,255,305,261,184,133,111,58))
```

Data (Dataset3)

```
list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(145,96,208,268,293,217,267,290,216,250,327,234,155,99,138,57))
```

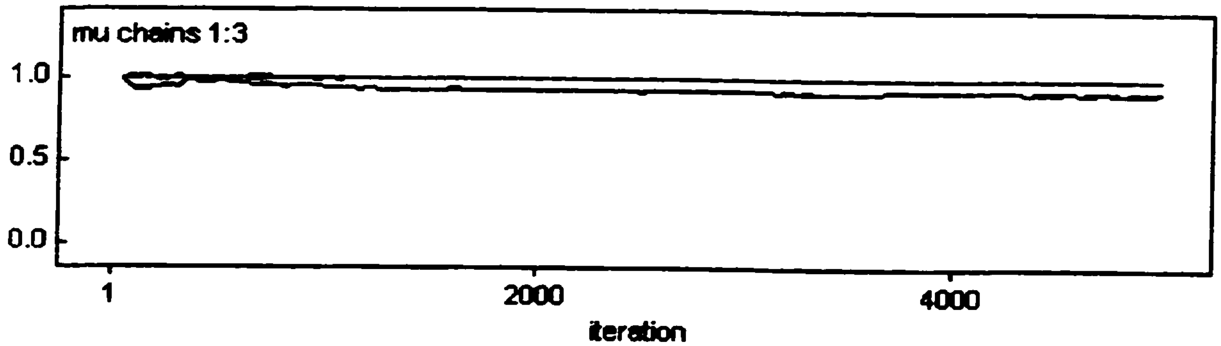
Inits(Chain1) `list(tau=1.0E-3,mu=0)`

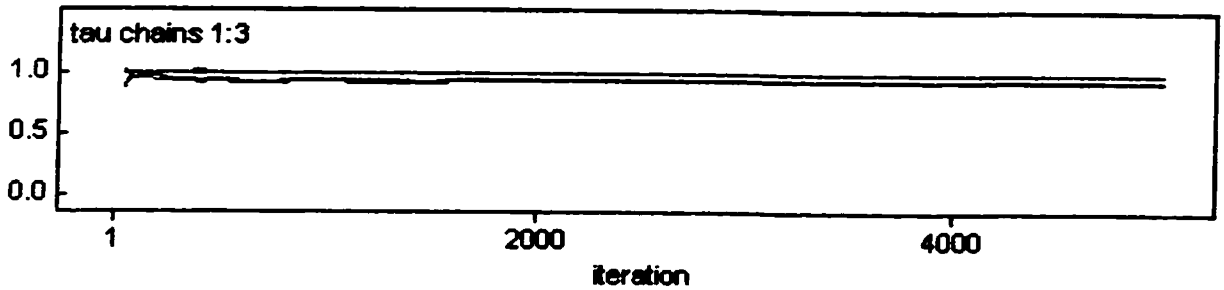
Inits(Chain2) `list(tau=1.0,mu=10)`

Inits(Chain3) `list(tau=1.0E-3,mu=10)`

Results(Dataset 1)

First, Gelman Rubin Diagnostic statistics to determine convergence. A burn-in period of 1000 iterations was used.





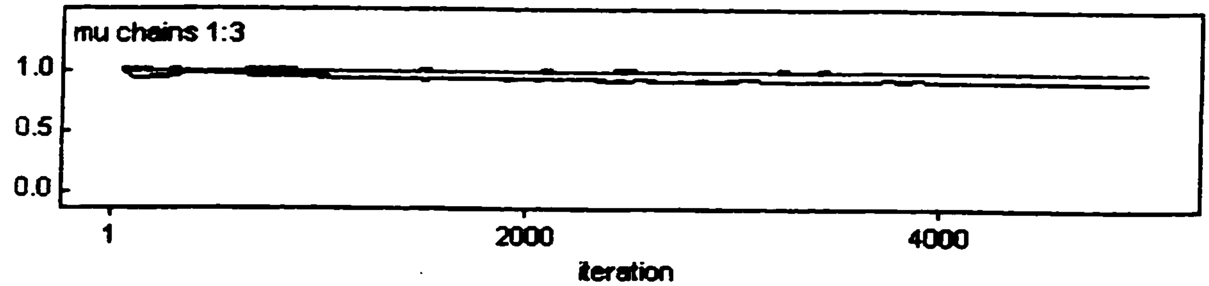
A 1000 update burn-in followed by a further 14000 updates (over 3 chains) gave the following parameter estimates:

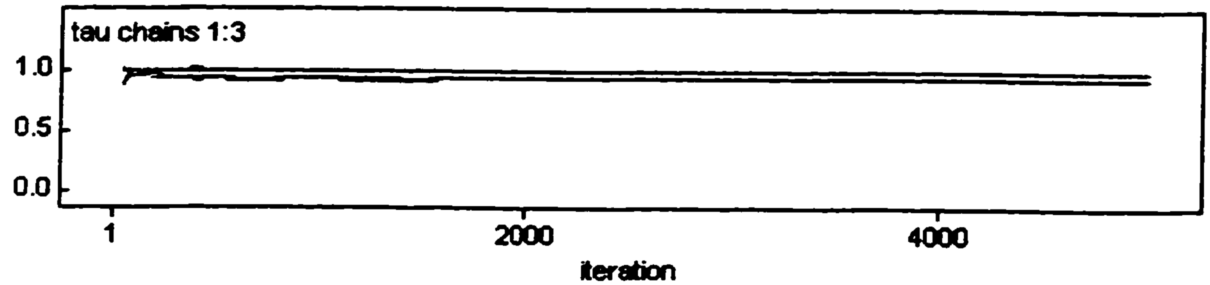
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
deviance	161.2	2.106	0.01686	159.1	160.5	166.9	1001	42000
mu	5.79	0.2984	0.002519	5.197	5.79	6.384	1001	42000
sigma	6.818	1.34	0.01113	4.773	6.623	9.991	1001	42000
tau	0.02388	0.008735	7.312E-5	0.01002	0.0228	0.04389	1001	42000

The underlying call length distribution in this example was Exponential(mu=6)
The raw (unaggregated) data had an actual mean of 5.79 and a s.d. of 5.89

Results (Dataset 2)

First, Gelman-Rubin diagnostics are calculated and plotted to determine convergence. A burn-in period of 2000 iterations was used.

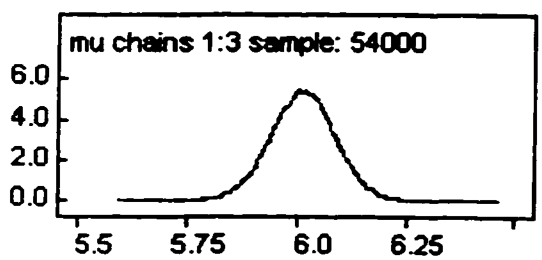


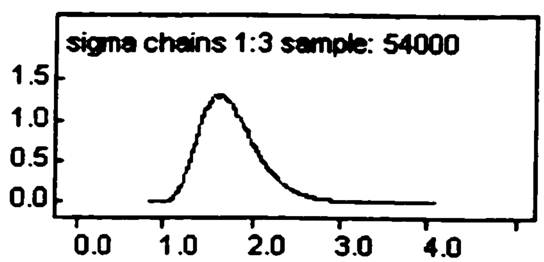


A 2000 update burn-in followed by a further 18000 updates (over each of 3 chains) gave the following parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
deviance	118.0	2.104	0.01542	115.9	117.4	123.7	2001	54000
mu	6.011	0.07727	5.714E-4	5.857	6.011	6.164	2001	54000
sigma	1.768	0.3475	0.002632	1.239	1.717	2.592	2001	54000
tau	0.355	0.1299	9.785E-4	0.1489	0.339	0.6517	2001	54000

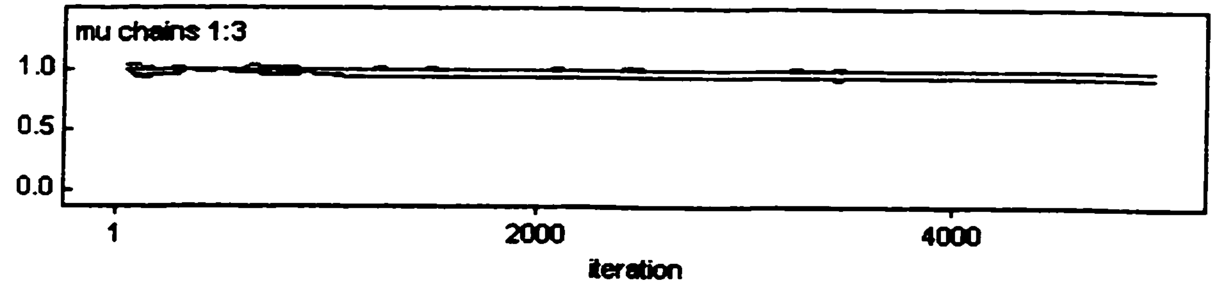
The underlying call length distribution (population) in this example was Normal(mu=6, sigma = 2). The raw (unaggregated) data had an actual mean of 6.01 and s.d. of 2.04.

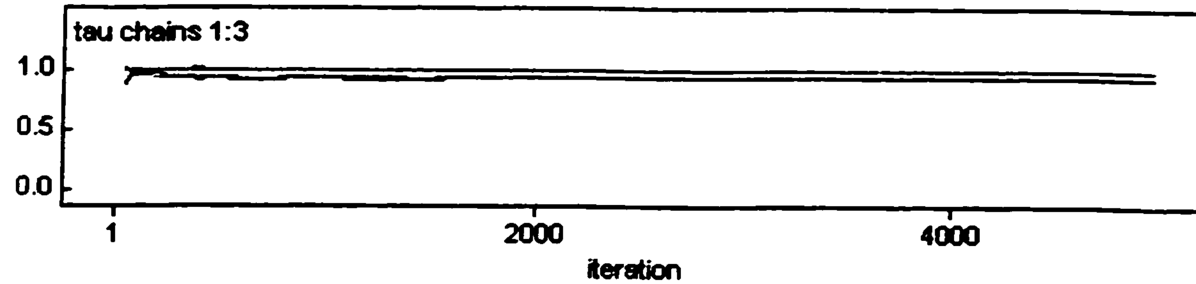




Results (Dataset3)

First, Gelman-Rubin Diagnostic statistics are calculated and plotted to determine convergence.





A 1000 update burn in followed by a further 14000 updates (over each of 3 chains) gave the following parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
deviance	149.6	2.106	0.01686	147.6	149.0	155.3	1001	42000
mu	6.035	0.2079	0.001756	5.622	6.036	6.449	1001	42000
sigma	4.751	0.9339	0.007759	3.327	4.615	6.963	1001	42000
tau	0.04916	0.01799	1.506E-4	0.02063	0.04694	0.09037	1001	42000

The underlying call length distribution in this example was Lognormal(mu=6, sigma=4). The individual call handle times had an actual mean of 6.04 and a s.d. of 4.29

Appendix A.3

WinBUGS Model based on Gamma Call Handle Times

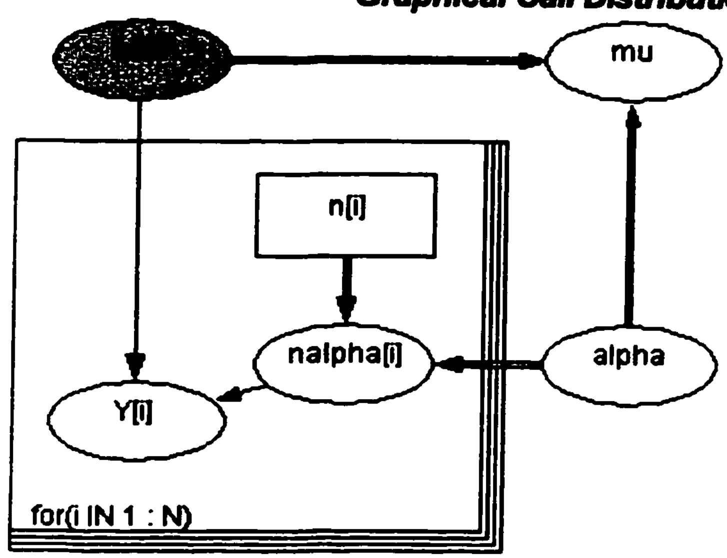
The model has been defined with the following distributions:

Y[i] ~ Gamma(n[i]*alpha, beta)

alpha ~ Gamma(0.001,0.001)
beta ~ Gamma(0.001,0.001)

alpha and beta are given independent, non-informative priors.

Graphical Call Distribution Model



Bugs Language for Call Distribution Model

```
model;
{
  for( i in 1 : N ) {
    nalpha[i] <- n[i] * alpha
    Y[i] ~ dgamma(nalpha[i],beta)
  }
  alpha ~ dgamma(0.001,0.001)
  beta ~ dgamma(0.001,0.001)
  mu <- alpha / beta
}
```

Data(Dataset1)

list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(146,95,210,265,186,191,299,364,230,273,235,172,184,80,127,71))

Data(Dataset2)

list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(123,119,166,237,230,240,296,293,235,255,305,261,184,133,111,58))

Data (Dataset3)

list(N=16,n=c(20,20,30,40,40,40,50,50,40,40,50,40,30,20,20,10),
Y=c(145,96,208,268,293,217,267,290,216,250,327,234,155,99,138,57))

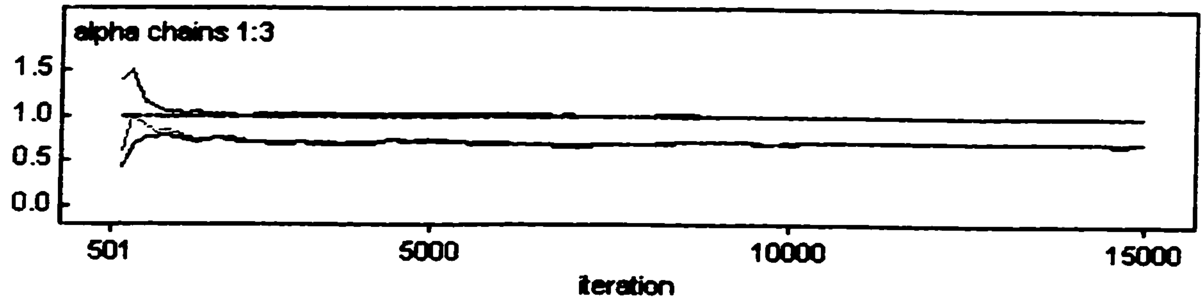
Inits(Chain1) list(alpha=1.0,beta=1.0)

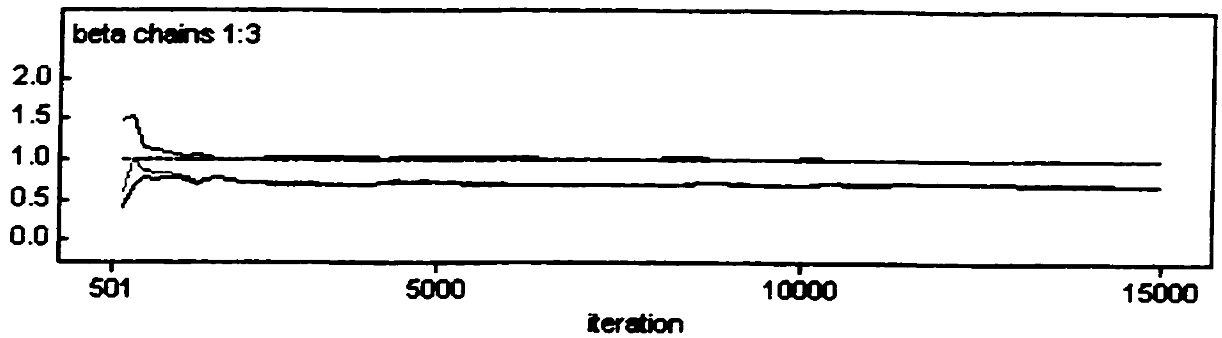
Inits(Chain2) list(alpha=0.10,beta=10)

Inits(Chain3) list(alpha=10,beta=0.1)

Results(Dataset 1)

First, Gelman Rubin Diagnostic statistics to determine convergence. A burn-in period of 10 000 iterations was used.





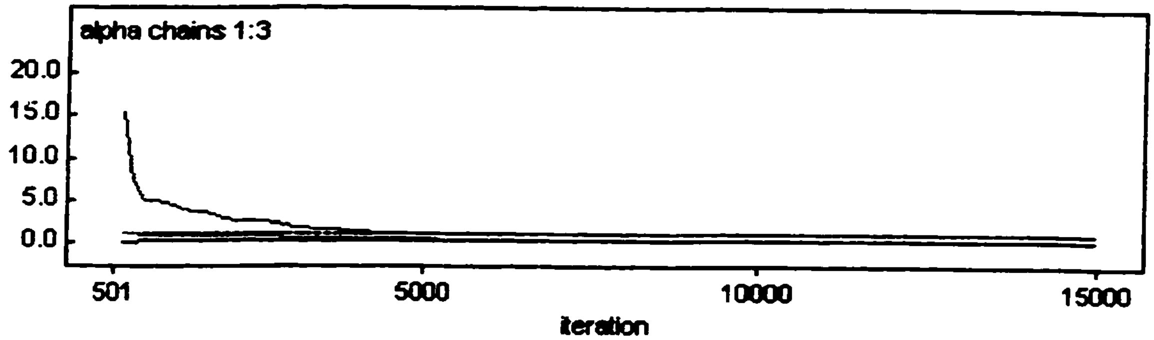
A 10 000 update burn-in followed by a further 20 000 updates (over 3 chains) gave the following parameter estimates:

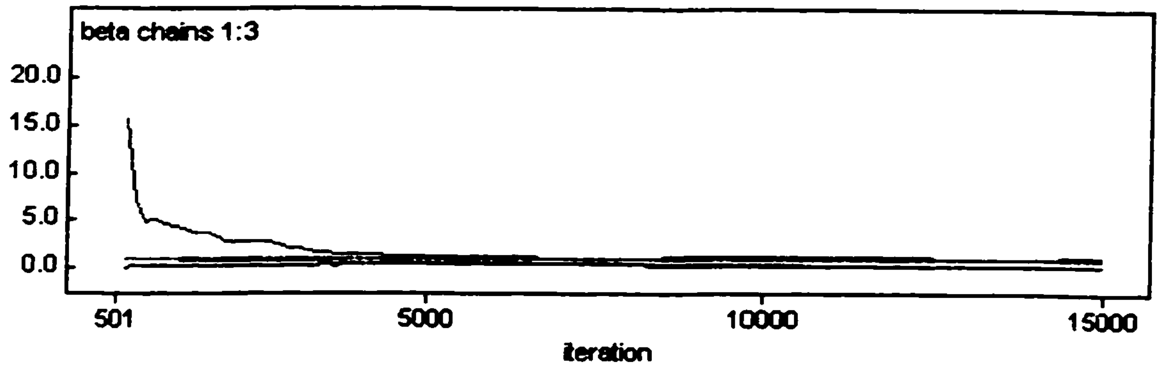
node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	0.753	0.2687	0.01107	0.3145	0.7243	1.359	10001	60000
beta	0.1299	0.04679	0.001926	0.05338	0.125	0.2351	10001	60000
deviance	161.6	2.172	0.05187	159.5	160.9	167.4	10001	60000
mu	5.812	0.3095	0.002364	5.237	5.801	6.458	10001	60000

The underlying call length distribution in this example was Exponential(mu=6)
The raw (unaggregated) data had an actual mean of 5.79 and a s.d. of 5.89

Results (Dataset 2)

First, Gelman-Rubin diagnostics are calculated and plotted to determine convergence. A burn-in period of 10 000 iterations was used.

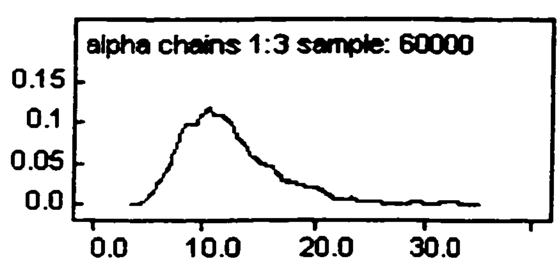


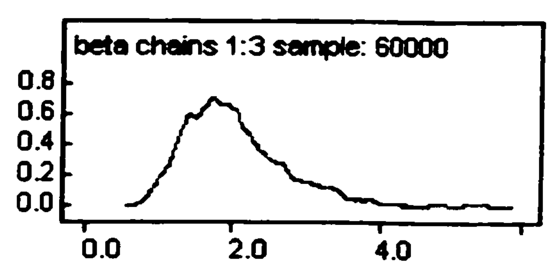


A 10 000 update burn-in followed by a further 20 000 updates (over each of 3 chains) gave the following parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	12.37	4.459	0.3485	6.148	11.56	23.24	10001	60000
beta	2.058	0.7421	0.058	1.022	1.923	3.867	10001	60000
deviance	117.6	1.924	0.09317	115.7	117.1	122.9	10001	60000
mu	6.013	0.07748	5.198E-4	5.86	6.012	6.167	10001	60000

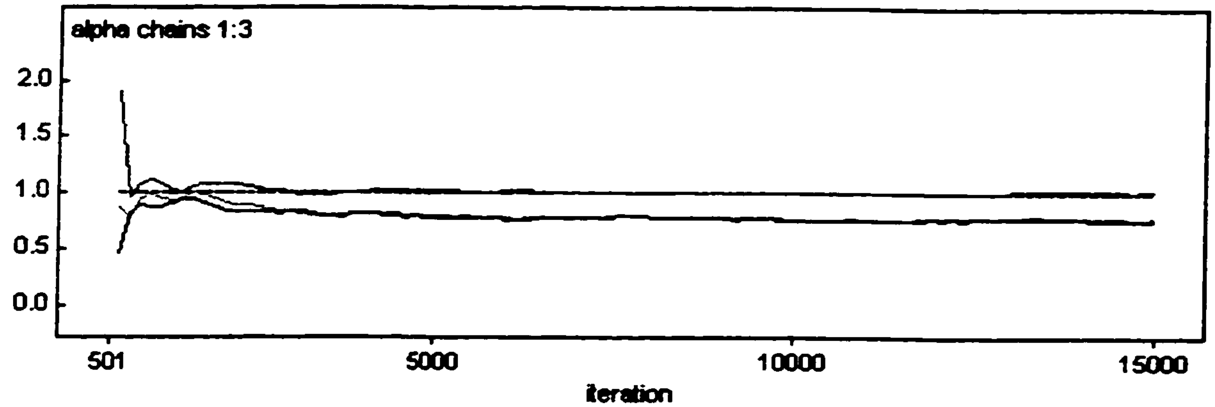
The underlying call length distribution (population) in this example was Normal(mu=6, sigma = 2). The raw (unaggregated) data had an actual mean of 6.01 and s.d. of 2.04.

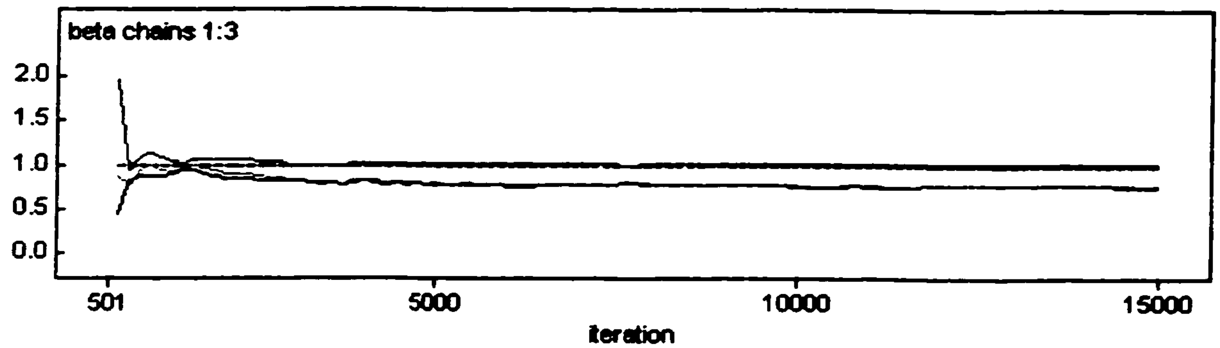




Results (Dataset3)

First, Gelman-Rubin Diagnostic statistics are calculated and plotted to determine convergence.

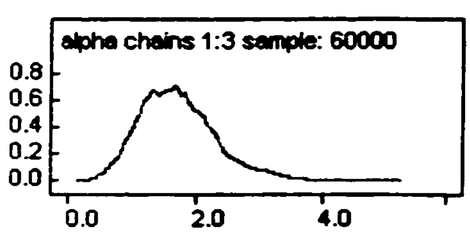


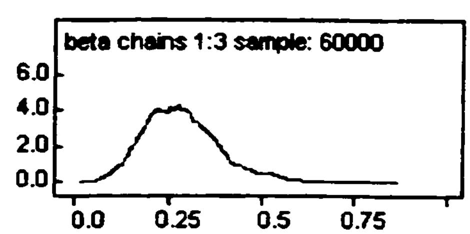


A 5000 update burn in followed by a further 20000 updates (over each of 3 chains) gave the following parameter estimates:

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
alpha	1.723	0.6046	0.03166	0.7168	1.666	3.129	5001	60000
beta	0.2854	0.1005	0.00526	0.118	0.2759	0.5194	5001	60000
deviance	149.2	2.205	0.06707	147.1	148.6	155.2	5001	60000
mu	6.046	0.2128	0.001555	5.641	6.041	6.483	5001	60000

The underlying call length distribution in this example was Lognormal(mu=6, sigma=4). The individual call handle times had an actual mean of 6.04 and a s.d. of 4.29

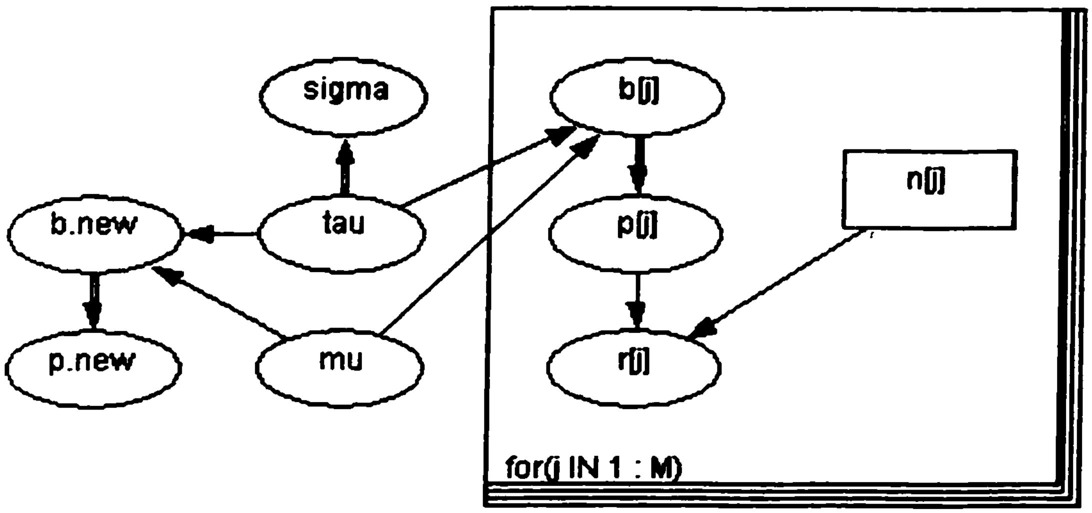




Appendix A.4

WinBUGS Model of Outbound Call Center Example

A binomial logistic regression is used to determine the success rate of new outbound call center campaign.

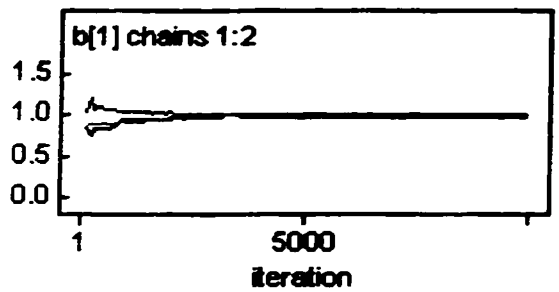


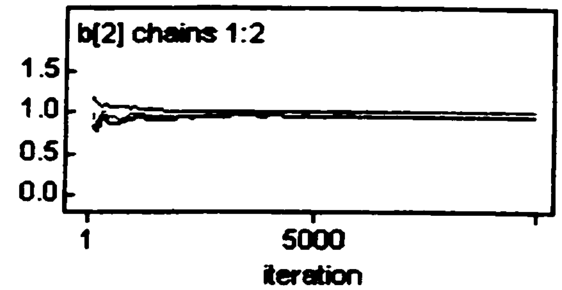

```
model;
{
  for( j in 1 : M ) {
    r[j] ~ dbin(p[j],n[j])
    logit(p[j]) <- b[j]
    b[j] ~ dnorm(mu,tau)
  }
  tau ~ dgamma(0.001,0.001)
  mu ~ dnorm( 0.0,1.0E-6)
  b.new ~ dnorm(mu,tau)
  logit(p.new)<-b.new
  sigma <- 1 / sqrt(tau)
}
```

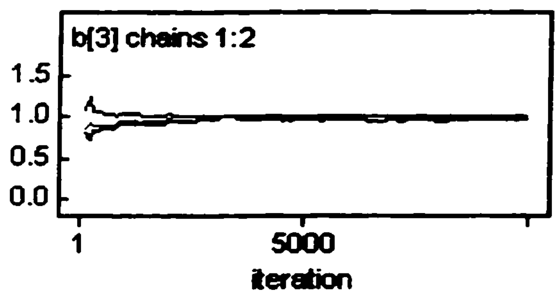
Data list(r = c(18, 9, 26, 13), n = c(30, 20, 44, 25), M = 4)
Inits1 list(tau=0.000001, mu= 10)
Inits2 list(tau=0.001, mu= -10)

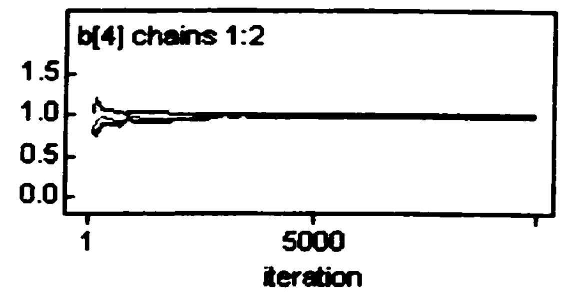
Results

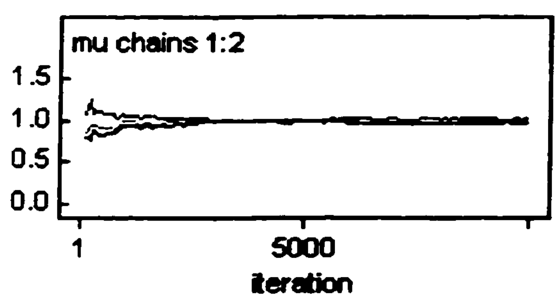
Using Gelman-Rubin statistic, it appears that we can assume convergence after 4000 iterations...

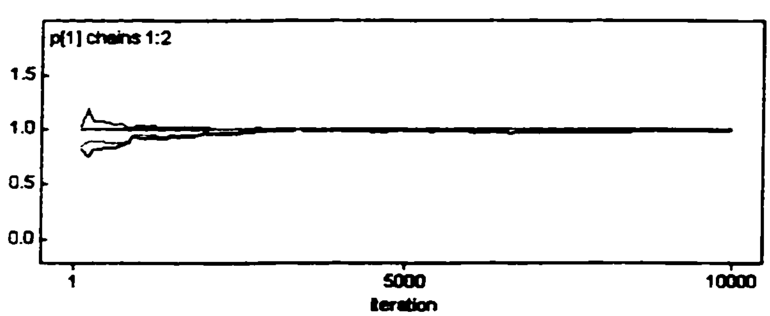


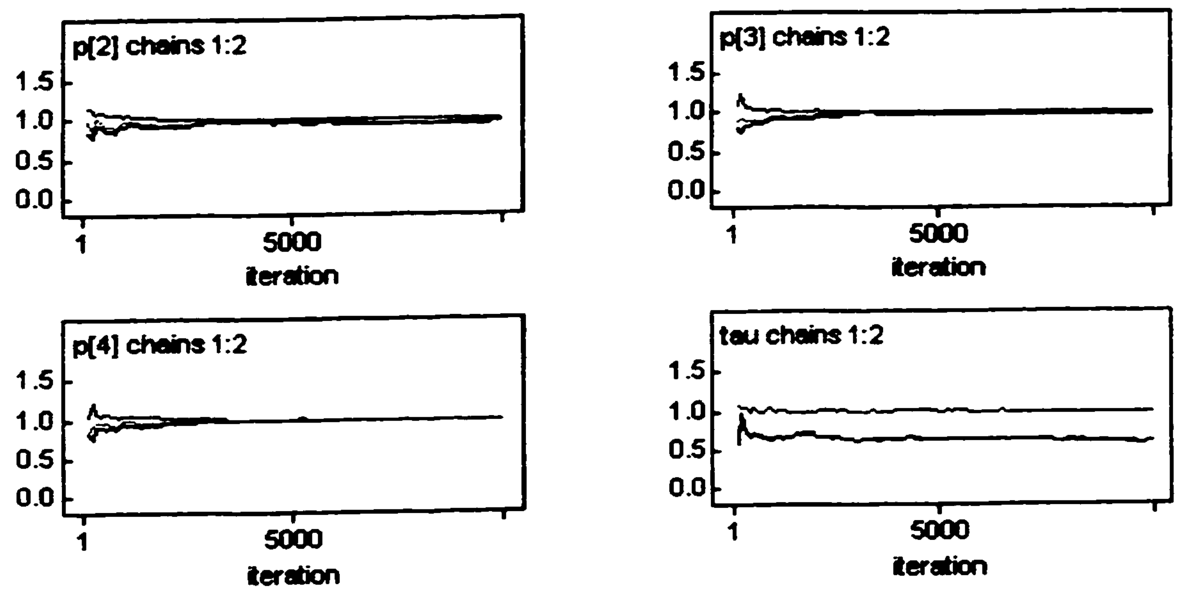




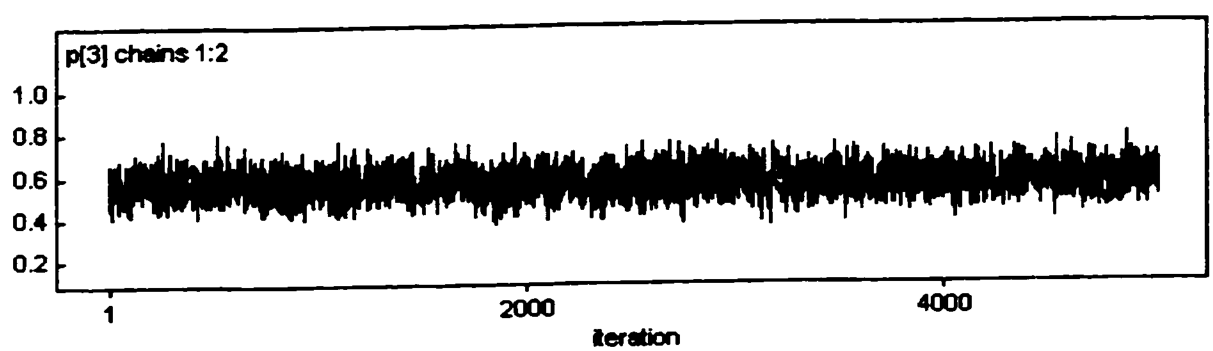








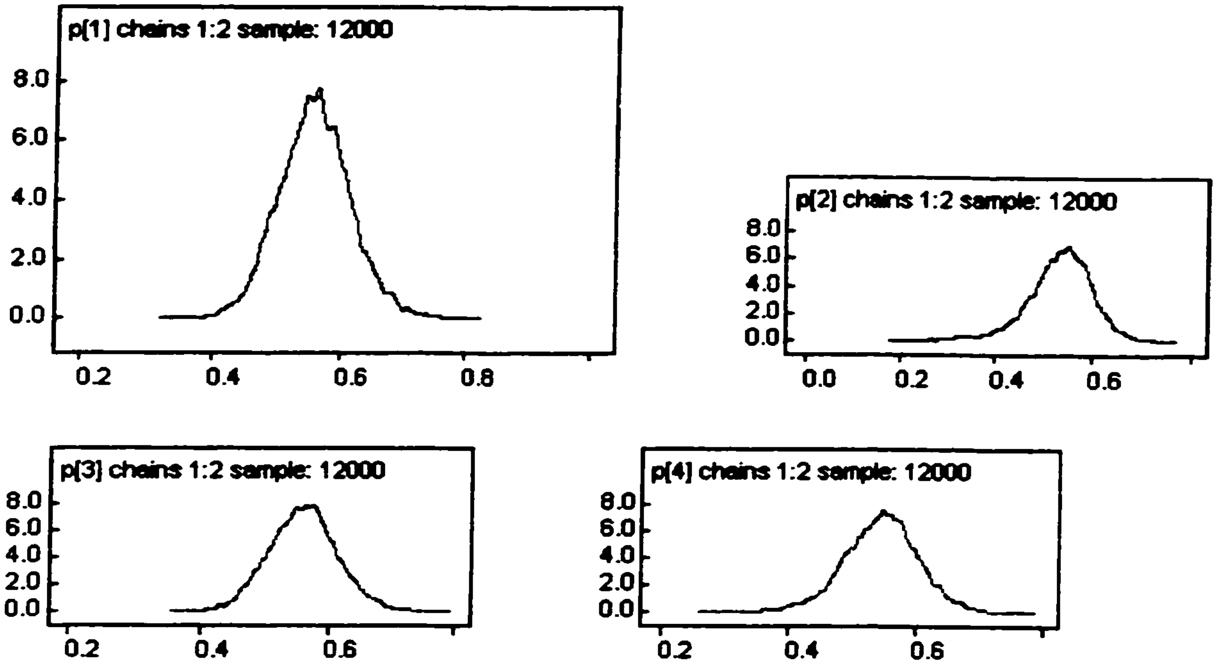
Also may want to monitor trace of two chains to confirm convergence and mixing.



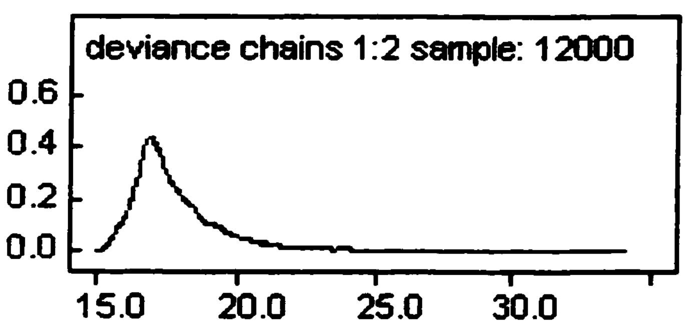
Statistics gathered for next 6000 iterations (2 chains). The parameter we are most interested in is p[i], the success rate of the new program at each outbound call center.

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
b[1]	0.265	0.2432	0.006517	-0.2072	0.2604	0.7693	4001	12000
b[2]	0.1476	0.2787	0.007378	-0.4946	0.1724	0.6301	4001	12000
b[3]	0.2662	0.2239	0.006217	-0.1625	0.262	0.7305	4001	12000
b[4]	0.1954	0.2534	0.006784	-0.3344	0.2059	0.6631	4001	12000
b.new	0.2174	0.4206	0.007992	-0.589	0.229	0.9847	4001	12000
deviance	17.88	1.768	0.04002	15.78	17.39	22.66	4001	12000
mu	0.2183	0.2545	0.006984	-0.2738	0.2226	0.6842	4001	12000
p[1]	0.5649	0.05856	0.001583	0.4484	0.5647	0.6834	4001	12000
p[2]	0.5364	0.06787	0.001806	0.3788	0.543	0.6525	4001	12000
p[3]	0.5653	0.05416	0.001514	0.4595	0.5651	0.6749	4001	12000
p[4]	0.548	0.06174	0.001659	0.4172	0.5513	0.66	4001	12000
p.new	0.5527	0.08951	0.001834	0.3569	0.557	0.728	4001	12000
tau	198.8	396.6	8.588	1.305	53.61	1304.0	4001	12000

Kernel Density Plots for success rate (p) at each call center



Posterior Distribution of the log-likelihood of the observed data (deviance)



Appendix B.1 Summary of Simulated ACD Data

			Exponential $\mu=6$	Normal $\mu=6$, $\sigma=2$ *(truncated)*	Lognormal $\mu=6$, $\sigma=4$
time period	#calls	cum #calls	average call length	average call length	average call length
1	20	0	7.3	6.2	7.2
2	20	20	4.7	6.0	4.8
3	30	40	7.0	5.5	6.9
4	40	70	6.6	5.9	6.7
5	40	110	4.7	5.8	7.3
6	40	150	4.8	6.0	5.4
7	50	190	6.0	5.9	5.3
8	50	240	7.3	5.9	5.8
9	40	290	5.8	5.9	5.4
10	40	330	6.8	6.4	6.3
11	50	370	4.7	6.1	6.5
12	40	420	4.3	6.5	5.8
13	30	460	6.1	6.1	5.2
14	20	490	4.0	6.6	5.0
15	20	510	6.3	5.6	6.9
16	10	530	7.1	5.8	5.7
	n =	540			

Appendix B.2 Exponential Data Detail

Exponential 6

	Time Block 1	Time Block 2	Time Block 3	Time Block 4	Time Block 5	Time Block 6	Time Block 7	Time Block 8	Time Block 9	Time Block 10	Time Block 11	Time Block 12	Time Block 13	Time Block 14	Time Block 15	Time Block 16
1	5.89	6.50	11.30	11.41	3.46	25.62	7.24	7.71	4.70	7.16	5.70	12.19	7.55	4.11	14.65	29.26
2	4.27	1.98	0.75	8.27	17.93	3.39	6.96	6.16	1.21	0.95	0.15	4.69	22.32	8.99	1.03	17.79
3	2.18	6.00	18.10	9.55	20.90	0.57	6.94	2.29	1.44	12.92	2.36	6.97	16.20	1.35	4.07	1.72
4	0.79	5.93	4.52	4.89	2.61	0.33	1.14	6.81	1.11	0.44	3.17	4.15	1.53	4.64	8.22	7.37
5	8.96	15.47	7.49	27.85	3.71	3.28	0.12	0.14	0.44	20.84	1.70	4.43	3.47	1.72	5.20	2.19
6	7.14	11.33	4.61	0.79	2.08	0.68	4.50	10.92	10.11	0.31	5.29	1.83	5.99	0.36	4.90	3.86
7	15.06	2.08	6.61	30.22	5.16	0.62	2.45	10.99	2.19	10.48	10.22	0.76	2.91	2.13	3.28	0.12
8	2.54	0.37	5.75	20.48	7.00	8.99	1.58	5.38	3.81	5.69	2.66	1.37	3.06	0.25	4.55	3.57
9	2.94	13.20	10.08	3.74	2.59	3.80	2.60	8.59	0.38	9.77	6.59	6.44	15.26	0.04	7.90	4.93
10	14.83	1.94	4.33	3.28	1.00	8.93	6.06	5.08	11.88	0.27	3.11	3.04	5.53	0.78	2.74	0.48
11	4.51	2.19	6.19	3.10	6.88	3.57	3.83	2.32	0.85	2.25	2.05	1.19	4.49	9.07	1.14	
12	0.33	2.42	13.45	9.15	7.73	4.35	5.28	0.99	2.36	1.59	1.13	7.62	2.95	0.86	3.14	
13	17.73	0.21	10.63	4.08	1.39	0.03	0.51	32.53	0.45	18.16	6.57	4.27	3.44	1.20	0.70	
14	4.15	8.14	3.95	6.70	1.58	6.98	6.59	2.70	7.93	2.98	6.61	0.26	15.17	7.95	35.85	
15	20.05	6.24	0.53	1.08	6.32	1.28	0.78	8.66	2.15	22.29	1.10	6.94	1.70	9.89	10.88	
16	3.14	0.60	22.44	0.67	8.28	2.50	7.54	10.74	11.39	4.16	2.02	0.76	4.43	5.23	2.69	
17	13.89	0.83	3.79	6.57	4.60	7.76	14.79	11.97	0.47	1.02	7.69	13.24	1.71	10.71	6.79	
18	0.82	0.39	2.42	0.27	0.35	3.68	5.83	10.05	0.82	14.81	4.29	0.78	0.66	3.09	7.56	
19	2.28	6.30	1.77	7.77	3.11	8.43	6.89	3.57	0.18	1.53	3.52	4.14	4.27	3.11	1.21	
20	15.02	2.42	1.39	1.75	3.90	0.59	6.53	4.34	0.64	5.42	0.07	2.16	3.60	4.63	0.18	
21			6.63	4.31	2.77	15.68	3.82	1.29	8.95	0.72	6.87	2.77	24.61			
22			3.88	5.64	0.81	10.67	24.99	35.64	4.02	15.82	7.59	5.82	2.08			
23			3.03	0.85	2.25	5.78	2.91	17.45	14.83	10.42	0.53	2.93	5.11			
24			5.51	6.34	1.06	0.28	4.16	5.72	17.11	2.71	0.45	7.49	0.17			
25			8.26	11.33	0.73	2.10	0.53	4.13	2.91	7.79	19.49	4.43	2.75			
26			2.44	2.82	2.07	12.80	2.16	0.21	11.59	5.99	2.45	3.86	3.36			
27			21.86	6.92	2.81	6.80	3.44	8.95	0.94	4.95	7.13	7.01	4.97			
28			15.00	2.77	14.22	0.78	0.36	3.08	24.11	3.33	0.83	0.03	10.56			
29			3.22	1.71	0.91	8.14	20.60	4.50	1.32	0.63	0.17	4.08	4.30			
30			0.30	3.73	0.80	4.74	4.57	0.79	7.77	2.19	4.09	0.61	0.06			
31				2.81	6.58	2.46	7.63	0.03	10.88	1.42	0.74	2.75				
32				14.18	11.63	0.98	0.98	5.67	0.50	5.42	8.87	5.29				
33				7.71	5.77	0.24	3.61	3.33	9.59	6.75	3.45	7.83				
34				6.82	0.06	0.81	5.14	13.14	13.74	3.34	0.89	5.90				
35				2.37	7.84	1.33	3.10	24.94	4.09	2.50	14.13	0.33				
36				3.88	1.35	4.62	9.30	0.56	0.78	11.44	2.39	14.17				
37				1.25	0.45	2.18	12.31	2.27	1.76	8.67	0.38	4.11				
38				4.39	11.51	2.80	7.54	12.77	9.43	0.13	11.43	0.79				
39				5.83	1.87	5.38	0.63	2.01	9.47	6.87	1.80	0.13				
40				7.49	0.11	7.43	7.65	0.28	11.91	29.16	6.06	4.08				
41							9.22	11.45			0.50					
42							7.73	13.25			3.67					
43							1.26	4.79			0.01					
44							4.85	0.86			2.87					
45							2.23	5.55			3.78					
46							9.67	3.71			12.03					
47							1.60	6.92			19.91					
48							25.08	8.07			0.70					
49							4.02	1.54			6.16					
50							9.81	9.65			9.17					
sum	146	95	210	265	186	191	299	364	230	273	235	172	184	80	127	71
n	20	20	30	40	40	40	50	50	40	40	50	40	30	20	20	10
avg.	7.3	4.7	7.0	6.6	4.7	4.8	6.0	7.3	5.8	6.8	4.7	4.3	6.1	4.0	6.3	7.1

Appendix B.3 Normal Data Detail

Normal 6 2 *(truncated at zero)*

	Time Block 1	Time Block 2	Time Block 3	Time Block 4	Time Block 5	Time Block 6	Time Block 7	Time Block 8	Time Block 9	Time Block 10	Time Block 11	Time Block 12	Time Block 13	Time Block 14	Time Block 15	Time Block 16
1	3.83	8.32	4.30	4.18	7.73	6.27	5.98	5.13	5.87	4.37	2.28	9.47	4.21	5.33	2.94	6.61
2	4.66	4.62	5.68	5.36	2.04	6.41	3.59	2.27	7.11	5.95	4.24	8.98	3.73	4.08	6.02	5.93
3	6.65	6.05	8.51	7.81	3.81	4.32	3.19	8.04	7.52	6.17	6.10	6.03	5.87	8.21	6.00	6.05
4	7.60	5.23	5.81	7.23	4.66	6.65	4.07	5.45	6.25	3.64	5.08	7.77	5.60	7.74	2.78	3.39
5	8.04	6.76	9.48	5.05	7.37	3.62	7.77	6.21	6.81	8.66	9.50	6.73	8.00	7.38	5.54	5.88
6	6.57	4.87	4.99	1.50	6.92	4.23	6.38	4.46	6.26	6.71	5.69	8.23	8.12	6.39	7.72	5.09
7	5.33	5.57	3.34	9.88	4.65	10.51	8.72	7.65	7.44	1.29	6.94	9.11	3.84	8.70	6.22	9.01
8	6.48	9.23	7.38	2.82	6.11	2.83	8.96	5.25	7.06	4.91	6.57	8.09	9.12	7.07	8.83	3.54
9	6.03	2.98	5.48	8.37	3.37	3.65	8.50	4.68	5.19	4.35	4.51	3.02	7.89	5.72	4.74	7.83
10	4.94	5.88	5.67	6.29	7.76	4.87	5.41	5.16	5.05	5.15	0.19	7.32	6.38	5.99	8.00	4.59
11	8.63	6.29	5.61	6.64	2.91	7.87	8.00	4.49	9.32	7.46	3.85	8.76	8.84	5.10	9.02	
12	7.81	6.84	5.15	10.60	7.89	5.02	6.06	5.59	3.13	3.97	0.84	5.13	6.64	8.13	7.68	
13	5.44	2.43	4.83	5.22	8.98	9.20	3.75	5.63	6.88	2.82	7.93	5.26	4.25	5.34	4.28	
14	7.29	4.92	4.18	6.06	6.26	8.08	11.05	5.27	0.00	6.62	3.63	6.78	6.35	5.46	4.37	
15	7.35	5.89	5.36	6.98	4.15	5.33	4.78	6.83	5.49	3.66	9.55	9.54	6.09	7.23	5.94	
16	8.24	4.46	7.10	5.83	6.33	8.97	4.76	7.45	5.54	8.37	2.06	2.97	6.71	6.94	2.87	
17	2.91	2.53	2.66	7.54	7.02	6.95	6.42	7.78	9.18	3.71	6.47	11.19	7.12	6.93	4.16	
18	5.44	8.29	4.26	3.44	6.37	6.84	3.87	7.03	5.88	9.14	7.93	5.85	5.15	7.06	6.29	
19	7.17	11.28	5.69	6.97	4.15	6.40	6.47	5.24	7.43	5.28	8.85	2.31	4.89	7.48	2.60	
20	2.76	6.86	3.13	5.04	5.23	10.18	7.54	6.04	7.75	3.58	8.49	3.60	4.33	6.43	5.12	
21			5.56	7.70	6.17	5.44	3.72	7.62	4.44	8.05	4.71	6.79	4.13			
22			5.20	7.60	4.21	9.42	5.85	8.24	5.99	8.32	3.74	4.89	6.25			
23			6.59	8.89	0.76	4.25	5.54	6.32	1.89	6.53	6.77	8.48	5.81			
24			8.40	5.11	3.42	8.30	5.58	6.17	3.69	9.10	4.73	7.39	1.81			
25			3.95	8.30	7.38	4.49	8.38	6.81	7.68	8.08	7.20	4.39	9.04			
26			7.03	2.33	5.25	7.08	5.19	5.93	5.47	8.14	9.65	3.58	6.52			
27			5.65	5.35	6.59	5.51	5.29	4.66	7.24	9.38	9.18	7.79	4.71			
28			4.98	2.48	9.64	6.57	5.66	1.22	5.87	5.35	4.98	11.93	8.44			
29			4.66	4.26	2.54	7.28	7.28	5.46	5.04	7.17	6.48	4.60	6.66			
30			5.25	3.64	5.71	4.73	7.07	6.52	5.77	6.68	6.70	9.60	7.17			
31				8.86	6.31	4.78	4.84	8.36	3.44	8.24	4.52	6.37				
32				11.27	10.74	3.22	5.46	5.94	4.42	10.07	7.25	3.24				
33				7.77	8.39	5.12	7.11	1.78	6.20	6.71	6.01	3.76				
34				3.81	9.29	2.19	1.81	7.27	7.36	8.62	8.70	6.87				
35				5.61	2.64	8.15	4.65	6.59	10.46	5.52	5.98	10.69				
36				6.64	4.72	5.67	4.29	6.20	6.16	5.73	6.47	4.79				
37				4.55	9.08	5.83	3.81	6.98	2.11	5.09	7.36	5.54				
38				3.83	3.49	3.60	6.48	7.61	5.14	8.74	9.27	2.79				
39				0.12	4.11	3.75	6.15	6.19	6.80	7.30	5.61	3.23				
40				5.66	5.78	6.54	8.86	6.77	4.87	6.72	3.14	8.25				
41							2.32	6.50			5.43					
42							8.18	5.17			6.18					
43							5.71	5.33			5.79					
44							2.75	4.64			5.73					
45							7.69	4.44			8.38					
46							6.66	5.49			5.62					
47							4.06	9.95			6.89					
48							6.81	0.66			7.57					
49							6.73	4.62			6.35					
50							6.37	7.64			8.03					
sum	123	119	166	237	230	240	296	293	235	255	305	261	184	133	111	58
n	20	20	30	40	40	40	50	50	40	40	50	40	30	20	20	10
avg.	6.2	6.0	5.5	5.9	5.8	6.0	5.9	5.9	5.9	6.4	6.1	6.5	6.1	6.6	5.6	5.8

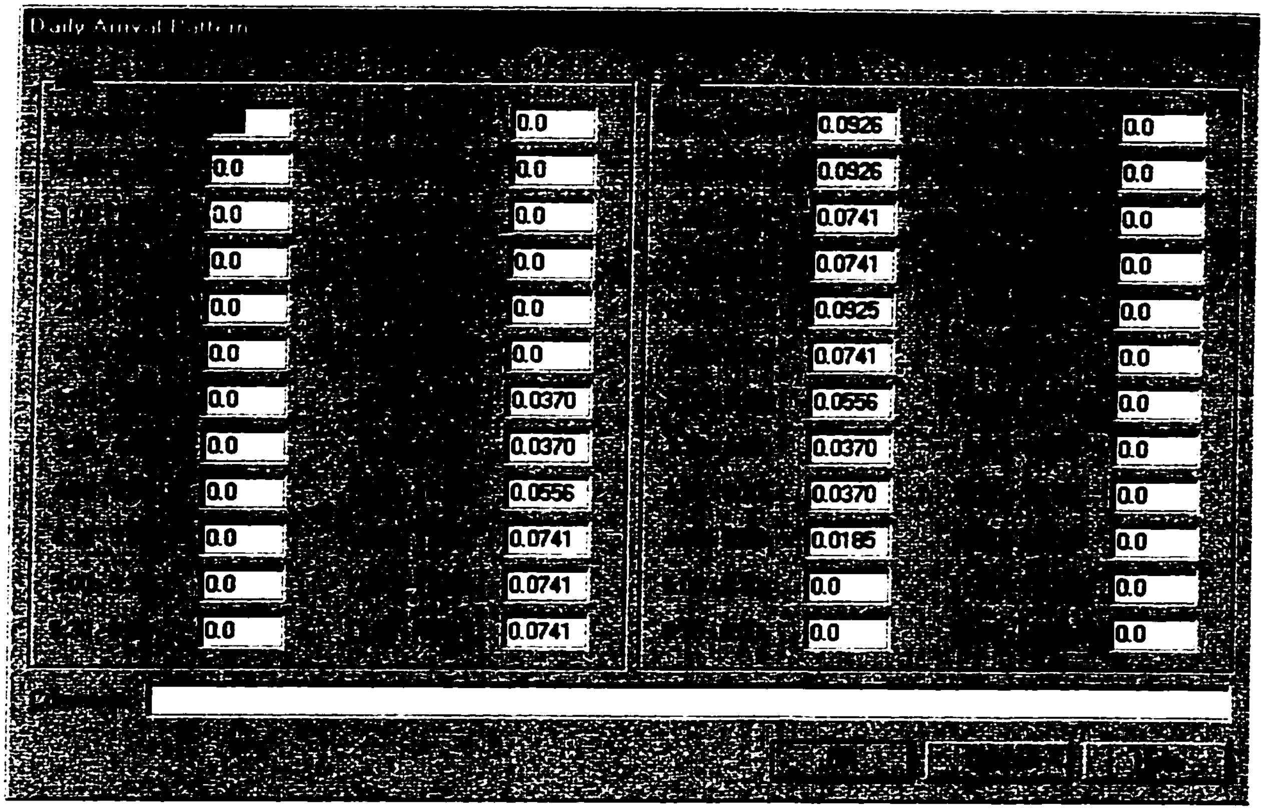
Appendix B.4 Logormal Data Detail

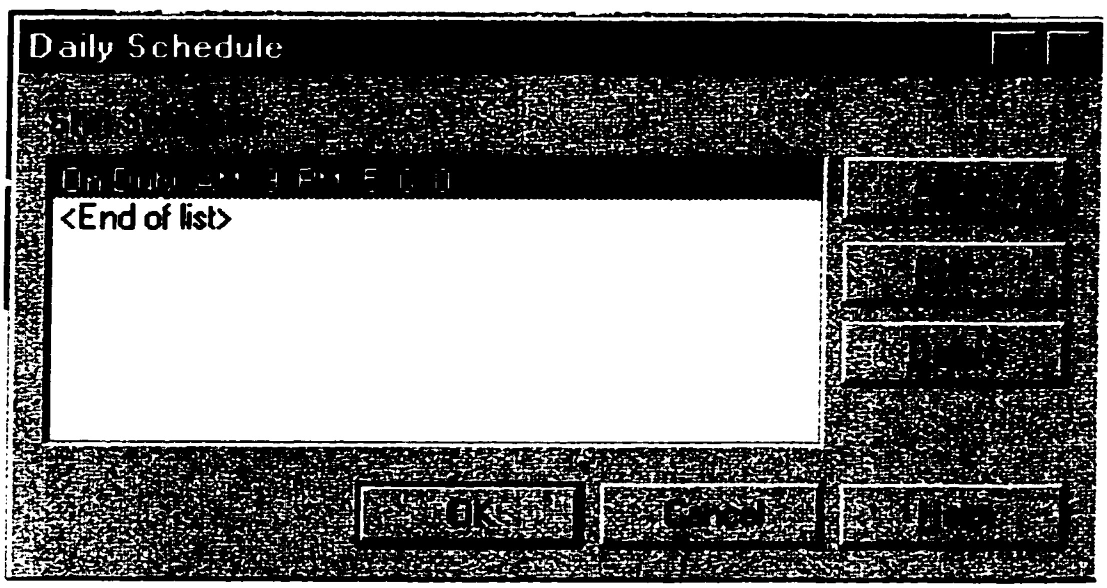
Lognormal 6 4

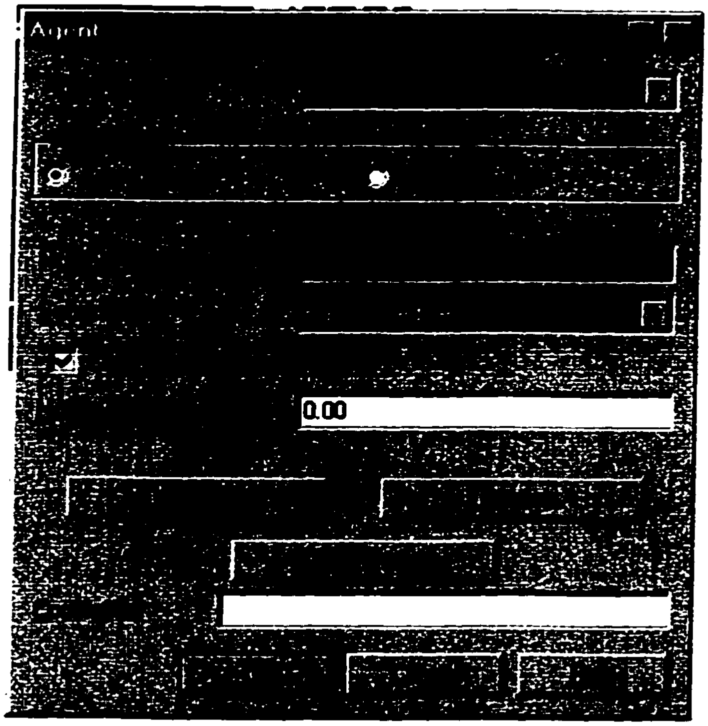
	Time Block 1	Time Block 2	Time Block 3	Time Block 4	Time Block 5	Time Block 6	Time Block 7	Time Block 8	Time Block 9	Time Block 10	Time Block 11	Time Block 12	Time Block 13	Time Block 14	Time Block 15	Time Block 16
1	12.83	3.75	1.98	4.09	13.88	4.82	11.26	5.01	2.45	2.60	11.12	5.50	3.36	2.72	9.25	5.88
2	4.01	5.34	17.60	5.70	6.05	7.63	21.74	5.28	3.48	3.05	4.62	5.14	2.78	3.41	3.05	3.81
3	1.82	3.78	0.82	2.76	4.49	1.35	3.70	6.97	3.06	10.99	4.69	3.72	3.63	8.57	2.96	5.02
4	4.24	2.74	7.89	1.34	2.61	9.86	2.85	1.04	5.91	4.02	4.29	10.21	4.94	3.43	9.08	5.39
5	2.18	6.23	5.44	2.87	5.83	2.51	3.42	3.37	5.92	4.65	8.38	9.58	3.79	3.19	5.14	6.53
6	1.40	4.73	8.09	6.45	3.20	1.84	4.03	1.54	5.38	3.08	2.82	2.44	8.69	3.63	21.36	2.74
7	14.17	3.90	2.31	5.51	12.73	9.37	2.82	12.97	2.88	6.92	8.57	8.44	4.90	4.04	3.98	4.00
8	8.34	3.78	7.62	43.91	15.94	7.37	5.10	1.50	2.73	15.13	4.84	2.84	4.00	2.90	1.98	6.71
9	5.21	3.53	5.74	2.81	6.64	5.59	4.27	9.99	4.85	1.29	3.35	2.11	7.23	7.08	3.40	10.24
10	5.82	5.73	16.54	4.88	1.94	7.32	1.33	5.72	2.98	7.43	1.44	11.84	4.18	5.69	23.39	6.56
11	4.16	2.80	3.80	12.34	5.68	1.31	5.02	7.05	4.95	6.32	2.55	4.71	7.11	5.60	4.66	
12	3.59	6.66	6.36	9.93	3.68	3.55	4.27	4.91	11.92	4.12	11.49	8.88	3.09	9.12	3.65	
13	24.18	8.84	17.16	7.59	6.07	4.33	7.04	3.96	2.99	3.47	6.64	1.68	3.03	9.64	6.21	
14	4.55	5.62	3.78	5.70	20.77	4.31	4.10	1.63	5.50	4.73	3.87	4.19	8.59	5.89	2.62	
15	5.84	8.82	19.94	5.60	5.75	5.84	5.39	4.71	7.00	13.33	2.18	6.13	3.10	3.95	8.50	
16	4.80	4.20	2.71	6.44	3.44	7.15	3.10	23.27	5.57	18.55	2.91	6.26	8.75	9.22	10.18	
17	3.91	0.94	5.24	3.02	0.73	4.27	3.46	8.31	11.69	3.42	10.76	6.04	7.15	3.10	3.97	
18	7.58	4.63	4.78	11.17	1.93	4.62	7.98	5.01	3.78	10.76	5.25	5.33	5.61	4.25	5.79	
19	20.85	3.77	5.54	1.73	15.77	3.64	2.72	2.41	2.41	3.65	5.36	6.34	5.54	2.08	4.73	
20	5.05	5.91	6.83	4.45	7.18	8.46	6.31	2.56	4.73	4.67	15.95	13.54	5.12	1.41	4.10	
21			1.33	2.27	7.73	2.10	3.53	4.62	9.53	8.24	8.64	5.34	4.07			
22			5.04	8.01	2.82	4.03	0.98	5.43	6.49	4.22	7.24	5.55	5.40			
23			4.84	5.54	3.33	1.92	4.96	4.09	4.55	3.77	3.70	0.56	5.19			
24			11.66	5.05	9.26	7.02	7.76	4.75	5.85	5.22	17.97	12.45	4.29			
25			5.28	3.24	5.23	5.66	8.82	2.65	1.34	1.91	9.07	4.34	6.47			
26			4.29	4.82	3.68	8.82	1.46	7.47	2.82	3.54	3.01	3.62	3.60			
27			3.58	8.64	4.57	3.56	16.52	3.60	3.93	1.46	11.57	4.39	4.97			
28			3.13	19.10	5.94	7.62	3.18	4.92	7.69	9.08	10.00	4.48	5.33			
29			10.27	5.77	18.94	9.68	2.58	5.06	9.53	2.37	6.42	2.48	1.98			
30			8.58	10.58	3.91	5.32	3.25	8.31	5.81	5.00	7.89	7.09	8.56			
31				3.14	7.16	4.41	5.39	2.12	5.51	9.78	2.81	4.25				
32				2.31	3.19	3.39	10.30	13.71	2.56	13.83	8.09	7.39				
33				5.69	3.17	4.88	1.84	3.98	1.19	3.70	6.22	8.24				
34				2.34	17.64	3.22	3.55	5.38	3.71	4.08	3.52	7.35				
35				1.77	9.14	7.00	3.66	5.17	3.67	8.58	3.78	3.71				
36				4.66	8.19	4.82	6.60	1.95	2.70	4.85	10.19	3.90				
37				10.84	2.70	7.75	6.27	5.24	7.57	4.23	6.99	7.53				
38				4.23	5.62	9.29	6.68	1.40	8.16	16.39	18.58	3.35				
39				7.49	11.48	3.28	2.75	8.80	9.07	6.37	4.58	6.31				
40				3.80	15.44	8.46	4.06	3.70	13.71	1.60	9.86	6.60				
41							5.41	10.12			4.20					
42							5.06	2.62			4.90					
43							5.82	5.55			3.58					
44							3.07	1.56			2.58					
45							3.37	11.59			6.67					
46							4.38	26.65			2.56					
47							7.75	2.58			6.24					
48							7.08	2.53			3.81					
49							4.34	1.98			5.25					
50							6.74	5.08			5.93					
sum	145	96	208	268	293	217	267	290	216	250	327	234	155	99	138	57
n	20	20	30	40	40	40	50	50	40	40	50	40	30	20	20	10
avg.	7.2	4.8	6.9	6.7	7.3	5.4	5.3	5.8	5.4	6.3	6.5	5.8	5.2	5.0	6.9	5.7

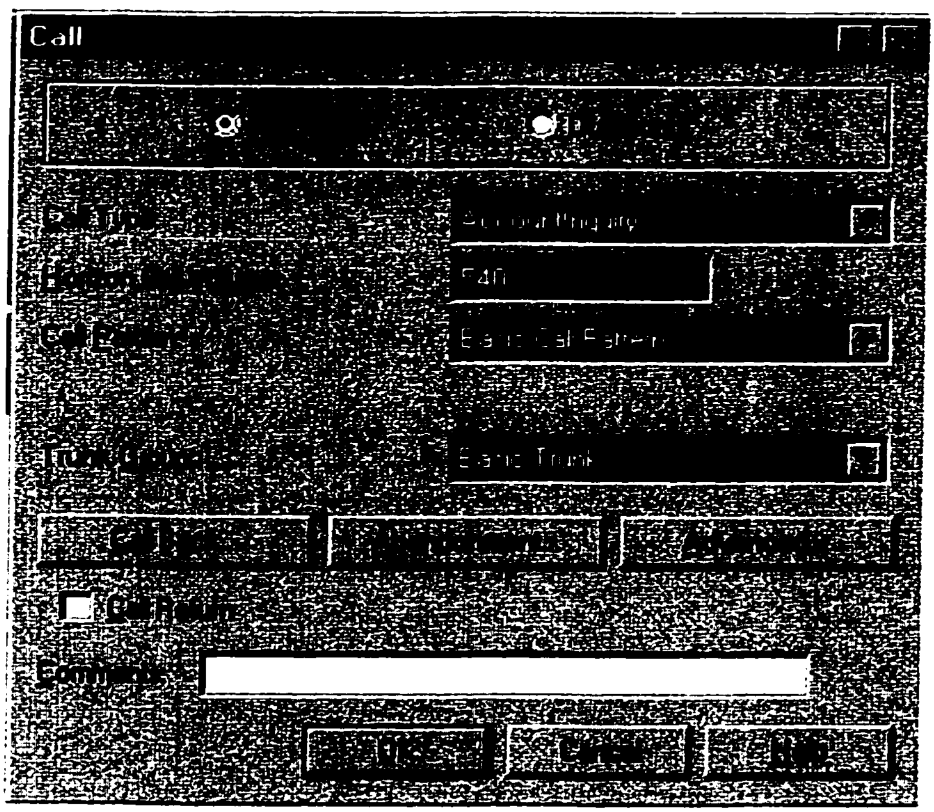
Appendix C

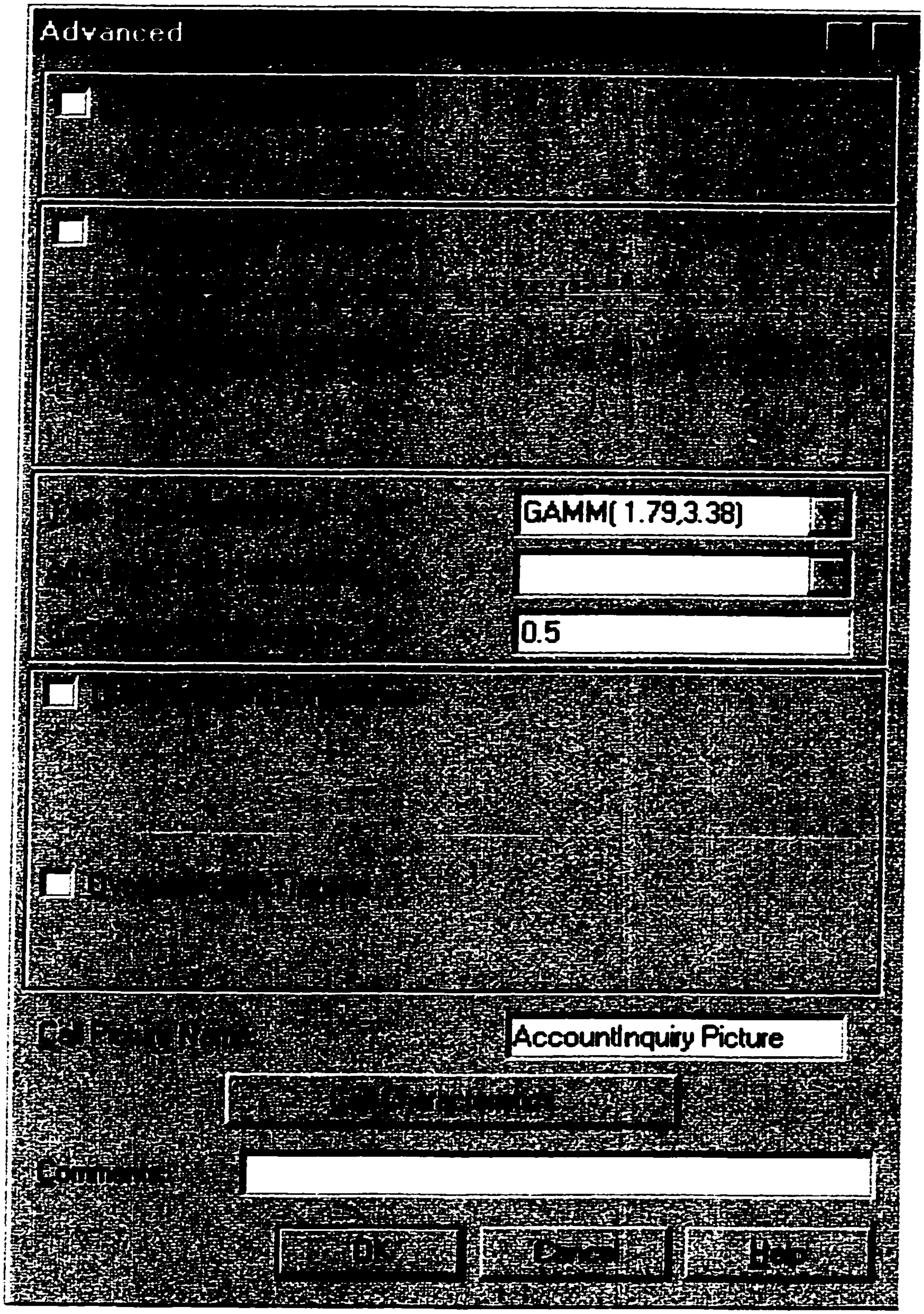
Screen Prints of Call$im Dialog Boxes






Agent
0.00


Call
Blank Trunk


Advanced
GAMM(1.79,3.38)
0.5
AccountInquiry Picture