Paper 94-2010

Using the JMP® 9 R Interfaces to Perform Bayesian Analyses: Why, How, and What?

Dave LeBlond, Abbott, Abbott Park, Illinois

Abstract

Modern computational algorithms such as Markov Chain Monte-Carlo (MCMC) make it possible to find exact solutions to a range of statistical problems that are beyond the reach of traditional statistical methods due to intractable mathematics or the requirement for over-simplifying assumptions. Adoption of these powerful approaches has been inhibited by a lack of appreciation of the Bayesian paradigm, as well as a lack of commercial software. Within the last decade, the advantages of these methods are becoming better known, and are now available in commercial software such as SAS.

With the inclusion of R interfaces in JMP version 9, a wide selection of statistical tools, including MCMC procedures, are now available to JMP users willing to learn R or WinBUGS language basics and write short JMP language (JSL) scripts. This paper provides simple, motivating examples of Bayesian estimation using random walk Metropolis or Gibbs sampling using JMP R interfaces. The examples include small sample coefficient of variation estimation and variance parameter and tolerance interval estimation for a two level hierarchical process. The examples are based on the author's experience in medical device and pharmaceutical R&D, but should be of general interest.

Background is provided so the reader can appreciate why using Bayesian/ MCMC approaches has value. The steps required to execute this kind of analysis are presented, including loading of the R and WinBUGS packages and required libraries, JSL scripts, and description of the associated theory. Finally, the output will be analyzed using JMP visualization and summary platforms.

Introduction

When a scientist, engineer, or statistician has a choice of statistical methods to apply to a critical problem it is generally best to use one that is familiar. This avoids learning curve delays and mistakes. However, anyone whose job involves extracting information from data is well advised to learn novel approaches to advance his/her skill level and versatility. In my role as a statistical consultant, there have been times when I have failed to heed this advice and instead, due to my ignorance, turned a client's problem into "one I could solve". At best this resulted in a confused client who could not interpret the statistical results provided. At worst, it was a disservice that resulted in a sub-optimal decision. So I recommend trying new techniques on example data when time permits.

The purpose of this paper is to share some good news with fellow JMP users: JMP version 9 includes an interface to the R environment for statistical computing. The R software includes many user contributed packages (2443 as of last count). Both the R software and packages are freely available. Of course, many of these duplicate statistical or graphical procedures that are already available in JMP. However, there are some R packages that permit statistical inferences that JMP cannot provide. We focus here on one particular kind of statistical inference that is now available to JMP users through R: Bayesian inference.

Most inferential procedures now available directly in JMP are based on sampling theory. Examples of sampling theory approaches include the t-, F-, and Chi-square tests that the reader is likely familiar with. By assuming an underlying population distribution (such as normal or binomial), we can predict the characteristics of data when sampling repeatedly from the population. We can use this sampling theory to obtain P-values for hypotheses tests, confidence intervals to contain underlying model parameters, tolerance intervals to contain future data, set specification limits to control a process, and address many statistical questions in a rigorous and objective way. During the 20th century, a plethora of sampling theory procedures were developed and they now form the core "tool kit" for much of our scientific and industrial decision making. What is most impressive about sampling theory is that it can do all this (much of the time) without making any prior assumptions about the true values of underlying distributional parameters (such as mean, standard deviation, or binomial proportion). In sampling theory, population parameters are regarded as fixed, but unknown, constants.

Bayesian thinking actually predated sampling theory by over 100 years. Like sampling theory, Bayesian inference presumes an underlying population distribution and treats observed data as the result of a random process. However, it goes one step further than sampling theory: It treats population parameters as random variables too. In Bayesian inference therefore, it is necessary to describe the "prior" distribution(s) of the population parameters as well as the

population distribution itself. The need to specify prior distributions seems subjective and undesirable. We often like to think of our statistical inferences about population parameters as being entirely based on the observed data. However, what I would like to illustrate here is that this apparent weakness of Bayesian inference, is in fact its most enduring strength. The need to specify prior distributions forces us to come to grips with critical assumptions; and as with the "Emperor's New Clothes" (ref 1), some problems benefit from a careful examination of underlying assumptions. Beyond that, the use of prior distributions gives us a knowledge building mechanism and allows us to address scientific and industrial questions that are beyond the reach of sampling theory alone.

Like any other form of data analysis, Bayesian approaches require some familiarity with new concepts, new languages, and attention to some important details. Bolstad (ref 11) is an excellent introduction to Bayesian ideas and basic methods. Gelman et al (ref 12) gives a much more in depth overview of Bayesian modeling and MCMC. Albert (ref 13) has provided a very nice LearnBayes R library and an accompanying text book filled with examples of Bayesian analyses in R. All of these resources are strongly recommended for JMP users who wish to build their knowledge of these important new tools.

To appreciate the "good news" aspect of JMPs new R interface, you first need to have some idea *Why* you would ever want to use a Bayesian procedure. You need to see *How* to use the R interface to make some simple, but I hope compelling, inferences, and finally you need to know *What* to be aware of when using Bayesian inference on your own. I will start with the *Why*.

Why take a Bayesian approach?

The pros and cons of Bayesian inference have been debated since the 1763 publication of Reverend Bayes' celebrated Essay (ref 2 and 3). Instead of joining that debate, I will simply offer some classes of problems for which I have found a Bayesian approach appropriate and useful.

Problems that require a probability statement about a model parameter

Bayesian thinking comes naturally to most decision makers who have little statistical training. Often their choices depend on their subjective knowledge of some unknown model parameter such as a defect rate. For instance, If the true defect rate for a lot is above an acceptance limit, they will fail the lot. They have some prior expectations about the defect rate and they have some lot testing data. They want to combine their prior knowledge with evidence from the data to know the probability that the lot defect rate is above the limit. Decision makers have much in common with gamblers. To make optimal decisions they need to know the odds. For this they often need probability statements about the underlying parameters of some process or lot.

A well trained statistician might apply a sampling theory approach to this problem and produce a hypothesis test or confidence interval estimate for the defect rate using only the available lot data. However, the statistician understands that inferences based on sampling theory cannot produce a probability statement about the underlying defect rate. A hypothesis test P-value of 0.04 does not mean that there is 96% probability that the defect rate is above the limit. The defect rate of the lot in question cannot be said to have a 95% probability of being contained within a 95% confidence interval estimated from the lot testing data alone. Instead, the P-value and confidence interval based on sampling theory allow statements about the statistical method itself and about the operating characteristics of the method when applied hypothetically in a repeated sampling sense. Such probability statements apply to the statistical method, not to the lot defect rate, and they may not be directly useful for probabilistic risk assessment. Often such statements are misinterpreted by decision makers.

On the other hand, a Bayesian estimate of the defect rate would come as a posterior distribution. The probability that the defect rate exceeds some limit is immediately available from the right tail area of this distributional estimate. Thus Bayesian approaches lend themselves to situations in which data estimates of model parameters are used to inform cost-benefit choices and risk assessments.

Problems that require a probability statement about some function of model parameters

Consider an analytical method whose measurement values are assumed to come from a normal distribution. The quantity of interest might be the underlying %CV which is defined as 100*sigma/mu. The objective might be to provide evidence that the method %CV is below some upper limit. Perhaps if the method %CV is above this limit, the method will not be fit for purpose and will require further development effort.

If the measurement values can be assumed to follow a normal distribution, sampling theory methods can be used to obtain an approximate confidence interval for the underlying %CV (ref 4). If the confidence interval upper bound is below the limit, the decision might be to approve the use of the method. However, as discussed above, this approach cannot answer the question "What is the probability that the method %CV is below the limit?". The confidence coefficient associated with a confidence interval derived from sampling theory refers to the hypothetical long term

repeated sampling performance of the statistical method used to get the confidence interval for the %CV, not to the underlying %CV itself. Therefore a decision maker cannot immediately use the confidence interval result to understand the decision risk in a direct probabilistic sense.

A Bayesian distributional estimate of the %CV can be used to answer this question directly. Further, the Bayesian approach would not require simplifying assumptions or complex analytical derivations. It can be applied to non-normal situations with little added complication. These comments apply as well to most cases in which a probabilistic statement is needed about any complex function of model parameters including those that involve random functions. The reason for this is that the distributional estimate of model parameters (i.e., mu and sigma) comes in the form of a sample from the joint distribution. This sample is similar in many ways to a bootstrap or Monte-Carlo sample and is exactly what is needed for simulations and predictions. The corresponding sample from the distribution of any function of the model parameters is available by simple calculation of the function for each draw. Many problems which are analytically intractable using sampling theory, such as the two sample t-test with unequal variances, become rather trivial using modern Bayesian computational tools. The new R interface makes such tools available to JMP users.

Problems in which relevant prior information is available

While a Bayesian approach does require that a prior distribution (sometimes simply referred to as a "prior") be placed on model parameters, one has considerable flexibility in the "information content" of the prior(s). The choice of prior is a subjective choice and members of a project team may disagree about the choice. The choice may have a large influence on an analysis if the prior information content outweighs that of the data. An analysis can be repeated with different priors to determine the impact this choice may have on the final conclusion. This can provide useful feedback to a project team.

The project team may choose to use a prior with low information content to let the "data speak for themselves". Priors can be chosen specifically to provide essentially no prior information about the value of a model parameter. For instance, as illustrated in Figure 1, one could use a continuous uniform prior between 0 and 1 as the prior for the true proportion defective in a population (say a large lot of "widgets").

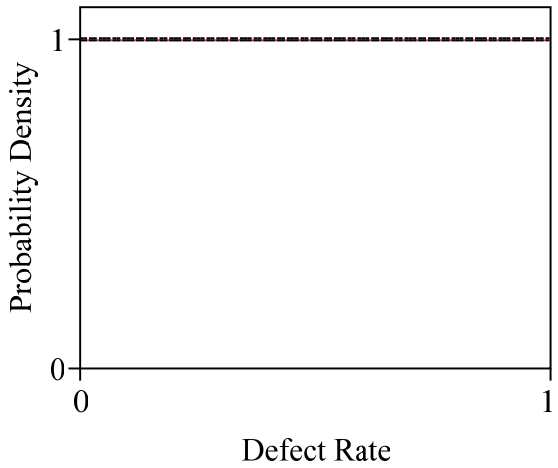

Figure 1. Non-informative prior distribution for the proportion defective in a process

The uniform distribution in Figure 1 can be constructed directly in JMP using the beta distribution function

Beta Density (Defect Rate, a, b) , [1]

where the first parameter is the hypothetical defect rate plotted on the horizontal axis, and the beta distribution parameters a = b = 1. All distributions have a "story" behind them. The story with the beta distribution in this case is that a + b widgets were randomly sampled and a were found defective. From this the information in the prior amounts to testing of 2 "prior" widgets and finding one of them defective. The resulting uniform distribution of Figure 1 is seen to give no preference to any particular value of the population proportion defective.

The result of a Bayesian analysis is always expressed as a "posterior" distribution of model parameters. Quite often, such a "non-informative" or "objective" prior will yield a posterior distribution whose quantiles are similar or even identical to that of a traditional sampling theory method confidence interval. To illustrate, assume actual testing of A + B = 30 widgets found A = 3 actual defectives. The posterior probability density distribution of the population proportion defective, given the non-informative prior of equation [1] is simply

Beta Density (Defect Rate, a+A, b+B) = Beta Density (Defect Rate, 4, 28) [2]

Which is plotted in Figure 2.

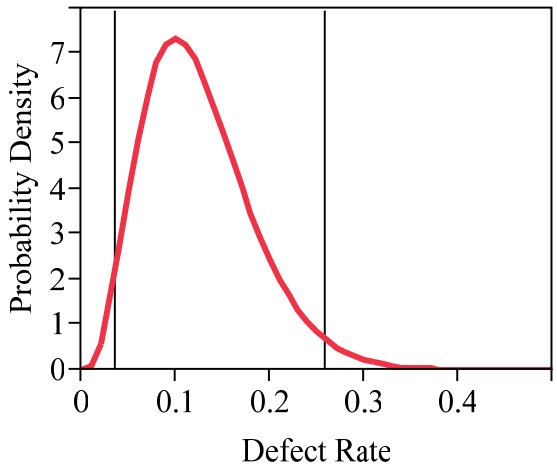

Figure 2. Posterior distribution for the proportion defective in a process based on the prior distribution in Figure 1 and the data given in the text

The equal tailed 95% "credible interval" estimate can be obtained using the beta quantile function as:

(Beta Quantile (0.025, 4, 28), Beta Quantile (0.975, 4, 28)) = (0.0363, 0.2575).

These are shown as vertical lines in Figure 2.

The 95% confidence interval obtained using a sampling theory based approach in the JMP Distribution platform is given in Table 1 as (0.0346, 0.2562), which is a similar range. However, the interpretation of these 2 intervals is very different. The Bayesian result allows us to state that the defect rate is between 0.0363 and 0.2575 with 95% probability. The sampling theory result allows us only to conclude that, since we have used a method that captures the true mean with 95% probability on repeated use, we can indirectly infer that we are unlikely to be wrong in assuming that the population value is between 0.0346 and 0.2562. The later statement is a probabilistic statement about the method, not the parameter of interest.

Table 1. Output from an analysis of a nominal variable consisting of the data for proportion of defects as discussed in the text.

Frequencies

Level	Count	Prob
0	27	0.90000
1	3	0.10000
Total	30	1.00000

Confidence Intervals

Level	Count	Prob	Lower CI	Upper CI	1-Alpha
0	27	0.90000	0.743789	0.9654	0.950
1	3	0.10000	0.0346	0.256211	0.950
Total	30				

Note: Computed using score confidence intervals.

For some parameters there may be considerable prior knowledge. Some prior knowledge is almost always available in product or analytical method development from relevant past experience, scientific literature, or applicable theory. For instance, in estimating a mean concentration by a well studied analytical method, there may be considerable prior knowledge about the method precision. Sampling theory approaches "waste" this information. However a Bayesian approach can capture this knowledge as an appropriate informative prior distribution on the method variance. If there is general agreement among knowledge experts and decision makers on the choice of an informative prior, then the

estimate of the mean will be more efficient because it leverages prior information. The opportunity to leverage prior knowledge is one advantage of a Bayesian, over a sampling theory approach.

Continuing with our "widget" example, One familiar with the widget manufacturing process may feel it is reasonable to assume that the defect rate likely not to be above 0.5 (50% defective product). One might express this prior information as a hypothetical prior experiment in which a + b = 10 widgets were sampled and a=1 was found defective. This prior density may be captured using

Beta Density (Defect Rate, 1, 9) , [1]

Which is shown in Figure 3.

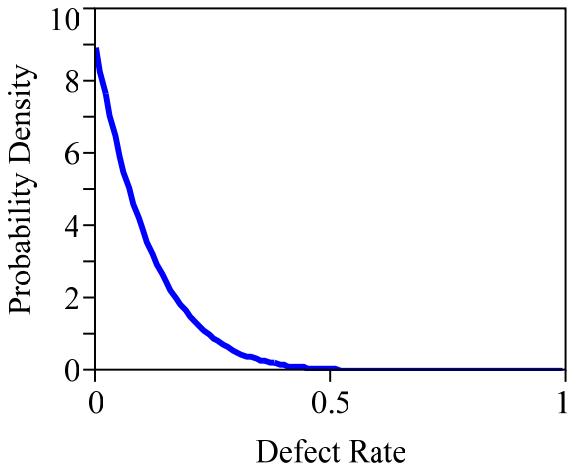

Figure 3. Informative prior distribution for the proportion defective in a process

Figure 4 gives the resulting posterior distribution (thick blue line) plotted using the following density obtained by setting a=1, b=9, A=3, and B=27. For comparison the densities in Figures 1 to 3 have been superimposed. Because the data set consisted of only 30 samples while the prior sample sizes were 2 and 10, the use of an informative prior causes the posterior distribution to be tighter with a lower mode than the posterior resulting from a non-informative prior. This reflects the information contained in the prior which, if justified, better informs the analysis and tightens the posterior distribution.

Beta Density (Defect Rate, a+A, b+B) = Beta Density (Defect Rate, 4, 36)

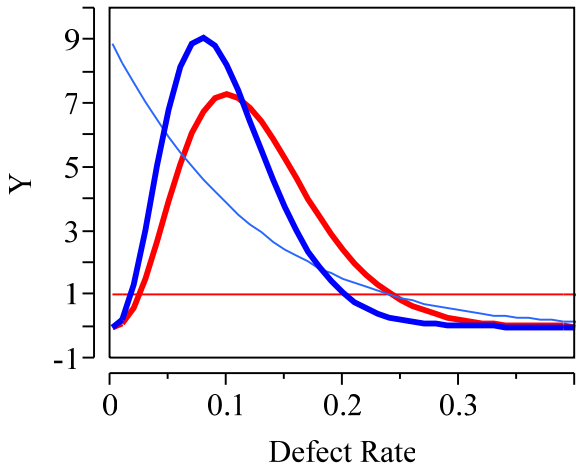

Figure 4. Illustration of the effect of prior assumptions on the estimation of a population proportion defective. The thin red and blue lines represent the non-informative and informative prior densities. The corresponding thicker lines give the respective posterior densities.

A Bayesian approach may be executed iteratively such that the current posterior distribution of model parameters becomes the prior information for the next experiment. This forms the basis for a rigorous knowledge building and continuous learning approach. As additional data are acquired, the relative importance of the original prior becomes negligible. The Bayesian approach is a continuous learning process that is in the spirit of much recent regulatory guidance in pharmaceutical development (ref 5 and 6). While it can be challenging to justify using prior information to regulatory authorities, successfully doing so has the potential of reducing the cost of developing new medicines.

Problems for which the data model involves a hierarchical structure

In some cases, a parameter value may be different for different sampling units. For example, the mean and/ or variance of the tablet drug level of a batch of tablets may differ among batches. In this example, there are 2 hierarchies: The between batch and the within batch level. In other cases there may be 3 or more hierarchies (e.g., tablets within bathes within manufacturing sites) and the internal structures can be endlessly complex. It is challenging to model such data structures analytically using sampling theory methods. Often some approximation, simplifying assumption, or invocation of a controversial principle is required to arrive at a solution. For instance, the confidence intervals for mixed model estimates produced by JMPs Fit Model platform are only approximate when sample sizes are small. Under some conditions, variance component estimates can actually be negative which is statistically embarrassing. Sampling theory concepts such as degrees of freedom, which are so compelling and straightforward in simple problems become slippery and counter-intuitive in the face of even the simplest of hierarchical structures.

Bayesian approaches are not as dependent on analytical solutions as are sampling theory approaches. Complex mathematical derivations and matrix algebra are replaced by a bottom up strategy in which the individual components of the data generation mechanism are each modeled probabilistically. Since each component can be thought of as conditionally independent (that is dependent only on the inputs from precedent components) the whole data generation structure can be simulated as the sum total of its components. In the last decade, Markov-Chain Monte-Carlo methods and their tendency to converge toward correct solutions after many iterations on our fast modern computers has given us good solutions to large classes of inference problems.

Because Bayesian solutions are derived by simulating the data generation process directly, the statistician (i.e., the computer programmer who writes the simulation “do loops”) gains considerable insight into the dynamics of the data generation in the model under consideration. Sometimes this insight is lost when using software procedure syntax or point-and-click platforms built on a distracting analytical solution somewhat removed from the data generation process.

Problems that involve missing data

Experimenters are well aware that experimental trials do not always yield usable results. In some experimental situations such as survival, reliability, or clinical trials missing results are the rule rather than the exception. Missing data can be conceptually challenging for sampling theory methods and the results of an analysis can depend on the particular missing data assumptions used in the analysis.

The Bayesian approach to missing data seems particularly straightforward. Missing values are simply treated as parameters to be estimated. As a Bayesian MCMC simulation progresses, the missing values are updated continuously along with all other parameters. The final Bayesian analysis produces predictive posterior estimates of the missing values.

Tests of equivalence

Statistical tests of equivalence arise frequently when a change to a manufacturing process or analytical testing method occurs. Such tests are used to establish the bioequivalence of two pharmaceuticals. Equivalence tests differ from tests of equality in that the null hypothesis is that the processes, methods, or pharmaceuticals are NOT equivalent. Such a test requires that an equivalence range be pre-defined for the parameters that govern the system. An example of a sampling theory based equivalence test is the two one-sided t-tests (TOST) test (ref 8).

As indicated above, confidence intervals derived from sampling theory give an indirect sense of the likely range of a parameter’s value. The only kind of probability statement possible with a sampling theory confidence interval is that associated with the long run repeated sampling performance of the method used to calculate the interval. Sampling theory concerns itself with sampling distributions of statistics only and does not provide, or permit, a probability distribution associated directly with ranges of parameter values. An equivalence test , on the other hand, is

concerned directly with ranges within which a parameter's value is likely to be. For instance the parameter may be the difference in the means or ratios of 2 populations whose equivalence is of interest.

To use sampling theory to make a test of equivalence, the following indirect procedure might be used:

1. State a range of parameter values within which the populations are considered equivalent.
2. Obtain an equal tailed 100*(1-2*alpha)% confidence interval for the parameter using a sampling theory method (such as a t-distribution based interval).
3. If the confidence interval is contained completely within the equivalence range, reject the null hypothesis that the populations are not equivalent with 100*alpha% confidence.

The evidence provided by this procedure is based on the long run performance of the sampling theory method itself. The final confidence level is not the probability that the populations are equivalent. In fact, the confidence level is often conservative and does not take into account the location or width of the final interval relative to the equivalence range. So the sampling theory approach does not permit an assessment of decision risk based on the data in hand.

The difficulty of interpreting the result of a confidence interval based equivalence test is exacerbated when the test involves multiple parameters. In this case, a multivariate confidence set must be identified in step 2 above. Multivariate confidence sets are not unique. They depend on choices such as shape. For instance, both rectangular and ellipsoidal confidence sets with the same confidence coefficient can be identified. Often the choice is arbitrary, but it may have a critical impact on whether the whole of the confidence region is contained within the equivalence region. While a multivariate equivalence test may still be conducted using multiple univariate confidence intervals for each parameter, it may be challenging to define the equivalence limits for each parameter if the multivariate equivalence region is not rectangular.

A Bayesian equivalence test is conceptually simpler and more direct. It consists of the following steps:

1. State a range of parameter values within which the populations are considered equivalent.
2. Obtain the (multivariate joint) posterior distribution for the parameter of interest.
3. Integrate the posterior distribution over the equivalence range. The result is the posterior probability that the populations are equivalent.

The Bayesian equivalence approach produces a probability that relates directly to the objective. There is no need to worry about the conservative nature of the test or non-essential issues such as confidence set shape. The resulting probability can be used directly to assess decision risk. The integration in step 3 is usually a simple counting exercise using the posterior sample provided by MCMC.

Adaptive studies involving interim analyses

Traditionally, clinical trials are prospectively designed. All aspects of the study are pre-specified and, except for safety monitoring, the protocol is not modified during the study. In fact, the data are not even examined by stake holders until the study is complete and the "blind is broken". To some extent, need for this rigidity is due to the nature of the sampling theory methods traditionally used to analyze the results and make the final decisions about safety and efficacy. In particular, stopping or changing a trial early based on partial data can be problematic and compromises the sampling theory interpretation of the results.

The Bayesian approach, on the other hand, leverages existing prior knowledge and builds that knowledge continuously over time (ref 9). This "learning as you go" paradigm encourages adaptive approaches such as progressively modifying the allocation of new patients to treatment arms likely to be safer or more efficacious. Bayesian approaches can lead to efficiencies by incorporating information from previous trials or from trials involving diverse patient demographics. While the Bayesian approach in drug trials is still controversial, its use in the approval of medical devices is well established (ref 10).

Predicting future performance

One of the motivations for modeling a process is to predict its future performance. A physician may want to predict the probability that a medical treatment will result in a cure – or produce an undesirable side effect. A manufacturer may want to predict the probability that a process will produce a product that fails release testing. Such events may have financial consequences and estimates of the associated probabilities support financial planning and cost/ benefit decisions.

Sampling theory provides exact prediction and tolerance intervals for simple problems. For more complex situations, approximations or computer simulation can be used to obtain the desired sampling theory intervals. The interpretation of these intervals retains a long run frequency interpretation. For instance, with a sampling theory prediction interval, a confidence level of 95% refers to the probability that, when applied over many studies, the method will successfully produce an interval which contains the next (or some other pre-specified, randomly selected) observation from the

study population. Similarly, a sampling theory based method that produces a 95% confidence tolerance interval to contain at least 99% of future observations, will produce an interval which actually does so in 95% of studies to which it is applied.

Prediction and tolerance intervals can also be constructed from a Bayesian point of view. However, the interpretation of the confidence level is different. For the Bayesian prediction (or tolerance) interval, the confidence level actually reflects the posterior probability that the interval will contain future values (or a specified proportion of future values). While the interpretation difference is subtle, the Bayesian interpretation lends itself more directly to decision making about the specific process under study. Exact Bayesian prediction and tolerance intervals are easily from the MCMC posterior sample for a wide range of complex problems which are difficult or intractable by sampling theory approaches.

How to take a Bayesian approach using JMP/R/WinBUGS

Users of JMP appreciate its excellent data management capabilities as well as its comprehensive set of analytical and graphical tools. For those who wish to operate from the JMP environment but add Bayesian options this section describes useful R interface commands, R libraries, and requirements for accessing WinBUGS.

Some Bayesian capabilities native to JMP

Below are listed some methods in JMP that are based on Bayesian thinking.

1. Box-Meyer Bayes Plot for identifying significant factor effects in saturated designed experiments.
2. BIC for model choice in D&I-optimal experimental design.
3. WeiBayes survival analysis - JMP can constrain the values of the Theta (Exponential), Beta (Weibull), and Sigma (LogNormal) parameters when fitting these distributions.
4. Bayesian variance component estimation in Variability can handle unbalanced data and forces all variances components to be positive and non-zero. The method computes the posterior means using a modified version of Jeffreys' prior. For details see Portnoy (1971) and Sahai (1974).
5. Bayesian information criterion (BIC) useful for model comparison.
6. For K-Means normal mixture clustering, JMP 9 has moved to a more stable algorithm, a Bayesian regularized version of the EM algorithm, which allows JMP to smoothly handle cases where the covariance matrix is singular.

These features are somewhat limited and JMP cannot perform MCMC, or other computer intensive functions that are needed for general Bayesian problems and which have been added to SAS. Currently if SAS is not available, WinBUGS and/or an R library may be good options to obtain a Bayesian analysis. At present, Bayesian methods and the supporting procedures are still not considered mainstream by the JMP development team. There may be a perception that such Bayesian methods are needed only by advanced experts. Perhaps as the advantages of the Bayesian approaches are more widely appreciated, they will move into mainstream use. In the meantime, the JMP development team has provided a useful interface to the R language. Within the R language, one has many Bayesian resources including access to WinBUGS. These are described below.

Downloading R and Winbugs

R and its associated libraries is available for free download from the following web site:

http://www.r-project.org/

The following R packages are recommended for Bayesian analyses: MCMCpack, LearnBayes, R2WinBUGS, and CODA. The graphics package lattice adds many graphical capabilities to R. These are conveniently downloaded from a CRAN site as zip files which can then be unzipped from within the R environment.

WinBUGS is also distributed free and can be obtained from the following web site:

http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml

Instructions for installing and using these packages are available on the web site or from the electronic manuals and HELP features that come with the packages. A discussion of the R and WinBUGS languages is beyond the scope of this paper. However, the HELP features of these packages

Accessing R using JMP Script

The following gives the basic JMP script structure for sending R code to the R environment from JMP.

```
R Init( );
R Submit("\[

… R code …

]\");
R Term( );
```

This basic structure will invoke R, execute the R code, and close the R connection. The last line above is needed only if you wish to close the R environment.

To make use of the R connection, it is necessary to send data to R. Lets say your data is contained in a JMP data file named dtname.jmp stored locally on your computer. You can send this data set to R using the following script:

```
dt = Open("…..\\dtname.jmp");
R Init ( );
R Send (dt);
R Submit("\[
.
.
R code. "dt" is now an R data frame and the dt column variables may be referenced in R
as dt$varname.
.
.
]\");
```

The R code indicated above is placed directly into the JMP script. This is where you would submit your instructions for a Bayesian analysis. Once an analysis in R has been completed, you will want to place your calculated variables (usually numeric vectors that may be posterior draws for a random variable of interest) into a hidden R data frame (say "df"), save this hidden frame as a file (say "dfname.jmp"), and then formally read that data frame into JMP so it can be properly viewed and referenced for further analysis and graphics. This can be done using the following script structure:

```
R Init( );
R Submit("\[
.
.
R code
.
var1 <-...
var2 <-...
.
df<-data.frame(cbind(var1=var1,var2=var2,…))
]\");
df = R Get(df);
df << save(".../dfname.jmp");
dt = Open("...dfname.jmp");
```

Once your data table is in JMP, you can manipulate it using the JMP GUI as desired. However, it may be convenient to add new columns (say "newcol") to the current data table (say referenced as "dt") using JMP script. This can be done using the following JSL, assuming newcol is a numeric result that is a result of some function you can define.

```
dt << New Column("newcol", Numeric, Continuous, Formula (…JMP formula to calculate
newcol…);
```

It is sometimes to change the modeling type of a variable (say "var1") from categorical to continuous in the currently selected data table. While this can be done manually, the following script may come in handy.

```
Column("var1") << SetModelingType("Continuous");
```

Accessing WinBUGS

While Bayesian analyses can be done directly in R using the MCMCpack, LearnBayes, or other R library, WinBUGS is a very comprehensive and flexible Bayesian analysis tool that reduces the programming burden somewhat. You can access WinBUGS on your computer through R with the R2WinBUGS library. While this may seem to be rather desperate (accessing R through JMP then WinBUGS through R), in actual operation it is relatively painless and can simplify the steps of performing steps in both WinBUGS and R separately. The JMP R interface lets you treat WinBUGS as a JMP resource. The following JSL structure shows a typical WinBUGS analysis.

```
R Init( );
R Submit("\[

# Create a data list where "vari" may be a scalar, vector, or matrix.
# This is produces the data set needed by WinBUGS
data<-list("var1","var2",…)

library(R2WinBUGS)

Modelname<-function(){
.
.
.
Place your WinBUGS model code in here
.
.
.
}
# Store the modelname as a local file
pathname<-file.path("…\\...\\","modelname.bug")
write.model(modelname,pathname)

# Define the starting values for the MCMC iterations
# If multiple chains are desired this may be a list of lists
inits<-list(var1=…;var2=…;…)

# Identify the random variables (nodes) for which the joint posterior sample is
desired
parameters<-c("var1", "var2",…)

# Place the WinBUGS output into an object called draws.sim
draws.sim<-bugs(data,inits,parameters,model.file="…\\modelname.bug",
             n.chains=…, n.iter=…,codapkg=FALSE,
             bugs.directory= "c:/Program Files/WinBUGS14/")

# Extract the posterior sample from draws.sim, Each chain must be separately extracted
# for illustration below chain number 1 ("chain1") is extracted
chain1<-data.frame(draws.sim$sims.array[,1,])
]\");

/* Move chain1 from R into a JMP data table */;
chain1=R Get(chain1);
chain1<< save("…/chain1.jmp");
Open("…/chain1.jmp");
```

What kinds of inferences can be obtained from Bayesian approaches: Some examples

The following examples will be discussed in greater detail in the presentation that accompanies this paper. In addition to the examples below, examples showing interval estimation for Cpk, estimation of tolerance intervals for complex cases, and nonlinear modeling will be illustrated. The author will be happy to provide the slides of this presentation when they are available. Please send enquires to david.leblond@abbott.com.

A variance component analysis using the JMP R interface to call WinBUGS

The dyes example is a famous example discussed by Box and Tiao (ref 14) and provided as an example in WinBUGS (see the HELP section in the software). The data should be present as a JMP data table in the File "Dyes stacked.jmp" containing 30 observations arranged as follows:

Batch	Sample	Yield
1	sample1	1545
1	sample2	1440
…		
6	sample3	1450
6	sample4	1480
6	sample5	1445

The following JSL script will move this data into WinBUGS then extract the posterior sample and produce some graphics that monitor convergence and give considerable insight into the inference beyond what would be obtained from JMPs Fit Model platform.

```
/* Open the Dyes data table */
Dyes=Open("C:\\Users\\Dave\\Desktop\\R2Winbugs Example\\Dyes stacked.jmp");

/* Analyze Using a Mixed model with random batch */
Fit Model(
	Y( :Yield ),
	Effects( :Batch & Random ),
	NoBounds( 1 ),	/* 1=unbounded 0=bounded */
	Personality( Standard Least Squares ),
	Method( EMS ),  /*REML or EMS */
	Set Alpha Level( 0.05 ),
	Emphasis( Effect Leverage ),
	Run(
		:Yield << {Analysis of Variance( 0 ), Lack of Fit( 0 ),
		Plot Actual by Predicted( 1 ), Plot Regression( 0 ),
		Plot Residual by Predicted( 1 ), Plot Effect Leverage( 1 )}
	),
	SendToReport(
		Dispatch(
			{"Response Yield", "Whole Model", "Actual by Predicted Plot"},
			"FitLS Leverage",
			FrameBox,
			{Grid Line Order( 4 ), Reference Line Order( 3 )}
		),
		Dispatch(
			{"Response Yield", "Whole Model", "Residual by Predicted Plot"},
			"FitLS Leverage",
			FrameBox,
			{Grid Line Order( 3 ), Reference Line Order( 2 )}
		)
	)
);


R Init( );

/* Send Dyes data set to R */
R Send( Dyes );

R Submit("\[
# Create the data set
y<-matrix(Dyes$Yield,ncol=5,byrow=1)
y
batches <- 6
samples <- 5
data<-list("batches","samples","y")

# Request the R2WinBUGS library (also requires CODA and Lattice)
library(R2WinBUGS)

# Specify model.file
dyesmodel <- function(){
		for( i in 1 : batches ) {
			mu[i] ~ dnorm(theta, tau.btw)
```

```
                    for( j in 1 : samples ) {
                          y[i , j] ~ dnorm(mu[i], tau.with)
                    }
              }
              theta ~ dnorm(0.0, 1.0E-10)
              # prior for within-variation
               sigma2.with <- 1 / tau.with
              tau.with ~ dgamma(0.001, 0.001)

              # Choice of priors for between-variation
              # Prior 1: uniform on SD
              #sigma.btw~ dunif(0,100)
              #sigma2.btw<-sigma.btw*sigma.btw
              #tau.btw<-1/sigma2.btw

              # Prior 2: Uniform on intra-class correlation coefficient,
              #             ICC=sigma2.btw / (sigma2.btw+sigma2.with)
              ICC ~ dunif(0,1)
              sigma2.btw <- sigma2.with *ICC/(1-ICC)
              tau.btw<-1/sigma2.btw

              # Prior 3: gamma(0.001, 0.001) NOT RECOMMENDED
              #tau.btw ~ dgamma(0.001, 0.001)
              #sigma2.btw <- 1 / tau.btw
}
# directory in which to (temporarily) store the model function
filename <- file.path("C:\\Users\\Dave\\Desktop\\R2Winbugs Example","dyesmodel.bug")
## R2WinBUGS function that creates the model file
write.model(dyesmodel, filename)
## and if you want to take a look remove the # in line below:
#file.show(filename)

# Initials depend on the model form. The following OK for 1 chain:
#      inits <- list(theta=1500, tau.with=1, sigma.btw=1)
       inits <- list(list(theta=1500,
tau.with=1,ICC=0.5,mu=c(1500,1500,1500,1500,1500,1500)),
                     list(theta=1000,
tau.with=2,ICC=0.8,mu=c(1000,1000,1000,1000,1000,1000)),
                     list(theta=2000,
tau.with=0.5,ICC=0.2,mu=c(2000,2000,2000,2000,2000,2000)))
#      inits <- list(theta=1500, tau.with=1, tau.btw=1)
#May also use the following forms (must change the parameter names)
# Specify inits
#inits <- function(){
#    list(theta=rnorm(J, 0, 100), mu.theta=rnorm(1, 0, 100),
#         sigma.theta=runif(1, 0, 100))
#}
## or alternatively something like:
# inits <- list(
#   list(theta=rnorm(J, 0, 90), mu.theta=rnorm(1, 0, 90),
#        sigma.theta=runif(1, 0, 90)),
#   list(theta=rnorm(J, 0, 100), mu.theta=rnorm(1, 0, 100),
#        sigma.theta=runif(1, 0, 100))
#   list(theta=rnorm(J, 0, 110), mu.theta=rnorm(1, 0, 110),
#        sigma.theta=runif(1, 0, 110)))

# Specify the nodes to save (parameters.to.save) Need to alter dep on model
parameters <- c("ICC", "sigma2.btw", "sigma2.with","theta")

## Below we call WinBUGS temporarily
## You may need to edit "bugs.directory"
## If codaPkg=FALSE, will get nice graph and posterior summary in R console
## If codaPkg=TRUE, the following will be saved to the directory tempdir()
##   codaX.txt where X=1,2,...,n.chains - this gives the posterior sample CODA file
##   codaIndex.txt which is the CODA index file
##   data.txt - the data file as a list
##   InitsX.txt - the initial values used to start each chain
```

```
dyes.sim <- bugs(data, inits, parameters,
    model.file="C:\\Users\\Dave\\Desktop\\R2Winbugs Example\\dyesmodel.bug",
    n.chains=3, n.iter=5000,codaPkg=FALSE,
    bugs.directory="c:/Program Files/WinBUGS14/")

# The following puts a nice stats summary of posterior & convergence in R console if
codaPkg=FALSE
print(dyes.sim)

# The following gives a nice plot of 80% credible intervals and Rhat for convergence
verification
# if codaPkg=FALSE
plot(dyes.sim)

# If codaPkg=FALSE, then the following will list the items available in the output
# See documentation for the bugs() statement to see what all these objects are
# In particular, sims.array contains the mcmc chains as a 3 way vector:
#  (draw number), (chain number), (parameter)
names(dyes.sim)
dyedraws <- data.frame(dyes.sim$sims.array[,1,])  #e.g. lists chain #1

#The following will start the coda interactive menu
#codamenu()
]\");

dyedraws = R Get(dyedraws); /* brings the MCMC draws into JMP as an invisible data
table */
dyedraws<<save("$SAMPLE_DATA/dyedraws.jmp"); /* saves the MCMC draws to a JMP data
file */
Open( "$SAMPLE_DATA/dyedraws.jmp" ); /* open the jmp data file as a visible data table
*/

/* The following plots the MCMC chain sequence and provides an autocorrelation
analysis*/
New Window( "dyedraws - Time Series",
       V List Box(
              Time Series( Y( :ICC ) ),
              Time Series( Y( :sigma2.btw ) ),
              Time Series( Y( :sigma2.with ) ),
              Time Series( Y( :theta ) ),
              Time Series( Y( :deviance ) )
       )
);

/* The following shows a 3D scatterplot of the draw sequence */
Scatterplot 3D(
       Y( :ICC, :sigma2.btw, :sigma2.with, :theta ),
       Connect Points( 1 ),
       Frame3D(
              Set Grab Handles( 0 ),
              Set Rotation( -72.0724159504983, 5.06473023667573, 38.6179709057463 )
       ),
       SendToReport(
              Dispatch( {}, "1", ScaleBox, {Max( 0.896009389671361 )} ),
              Dispatch(
                     {},
                     "2",
                     ScaleBox,
                     {Min( 683.229813664596 ), Max( 28183.2298136646 )}
              ),
              Dispatch( {}, "3", ScaleBox, {Min( 1128.75536480687 )} )
       )
);


/* The following gives kernal density estimates and credible intervals */
Distribution(
```

```
        Continuous Distribution( Column( :ICC ), Fit Distribution( Smooth Curve ) ),
        Continuous Distribution(
                Column( :sigma2.btw ),
                Fit Distribution( Smooth Curve )
        ),
        Continuous Distribution(
                Column( :sigma2.with ),
                Fit Distribution( Smooth Curve )
        ),
        Continuous Distribution( Column( :theta ), Fit Distribution( Smooth Curve ) ),
        Continuous Distribution( Column( :deviance ), Fit Distribution( Smooth Curve )
)
);

/* The following generate 2D kernal density estimates of some parameter pairs */
Bivariate(
        Y( :sigma2.btw ),
        X( :theta ),
        Nonpar Density( {Kernel Control( 1 ), Set Kernel( 11.068, 1191.7 )} )
);

Bivariate(
        Y( :sigma2.btw ),
        X( :sigma2.with ),
        Nonpar Density( {Kernel Control( 1 ), Set Kernel( 379.66, 1128.9 )} )
);

Bivariate(
        Y( :sigma2.btw ),
        X( :ICC ),
        Nonpar Density( {Kernel Control( 1 ), Set Kernel( 0.06228, 836.2 )} )
);

R Term( );
```

Obtaining a Credible and Prediction interval for a %CV

The %CV is often used as a measure of variation for measurement systems or observational data in which it is reasonable to assume that the standard deviation is proportional to the mean. Interval estimates for the %CV can be derived using sampling theory, but the equations are complex and usually only valid when larger sample sizes are available.

The following example uses the JMP R interface to obtain an exact Bayesian credible interval estimate for a small sample %CV using the R library MCMCpack.

```
R Init( );

R Submit("\[
#Credible interval for the normal CV by Metropolis random walk

# Data from Mark Vangel American Statistician 15(1) pp21-26
# Nearly exact conf interval by method of McKay: (2.070%, 12.93%)
y<-c(326,302,307,299,329)

# Obtain exact confidence interval
library(MBESS)
ci.cv(data = y,conf.level = 0.95)

# Hyper paramters (mu0,kappa0,nu0, sigma02)
# Parameterization of Gelman p78
hyper<-c(300,0,-1,0)

#Log Posterior Function
logpostnorm2=function(theta,y, hyper){
  # we want to parameterize on theta =(mu,ln(sigma)) here even though
```

```
  # the posterior is parameterized on (mu,sigma^2)
  # so must include a log Jacobian of +2*theta[2]
  mu0<-hyper[1]
  kappa0<-hyper[2]
  nu0<-hyper[3]
  sigma02<-hyper[4]
  # identify the model parameters
  mu <- theta[1]; sigma <- exp(theta[2]); sigma2<-sigma^2
  # Calculate sufficient statistics
  ymean<-mean(y)
  n<-length(y)
  s2<-var(y)
  # Obtain paramters of posterior
  mun<-(kappa0*mu0+n*ymean)/(kappa0+n)
  kappan<-kappa0+n
  nun<-nu0+n
  sigman2<-(nu0*sigma02 + (n-1)*s2 + kappa0*n*(ymean-mu0)^2/(kappa0+n)   )/nun
  post1<- dnorm(mu,mun,sqrt(sigman2/kappan),log=TRUE)
  post2<- log(dinvgamma(sigma2,nun/2,nun*sigman2/2))
  return(post1 + post2 + 2*theta[2])
}
# Need the MCMCpack
library(MCMCpack)

# Obtain the posterior mode and hessian at the mode
optimum<-
optim(hessian=TRUE,par=c(mean(y),log(sqrt(var(y)))),fn=logpostnorm2,y=y,hyper=hyper,
      method = "BFGS")
# Perform the MCMC
bayesfit<-MCMCmetrop1R(fun=logpostnorm2,y=y,hyper=hyper,V= -solve(optimum$hessian),
             theta.init=optimum$par,
             thin=1, mcmc=40000, burnin=500,
             tune=c(1.5, 1.5),verbose=0, logfun=TRUE,seed=1);

mu<-bayesfit[,1]
LNsigma<-bayesfit[,2] # Note the parameter is on log scale
bayesfitdf<-data.frame(cbind(mu=mu,LNsigma=LNsigma))

]\");

/* The below works as long as there is not already a file with this name in the
directory */
bayesfitdf = R Get(bayesfitdf); /* brings the MCMC draws into JMP as an invisible data
table */
bayesfitdf << save("$SAMPLE_DATA/bayesfitdf5.jmp"); /* saves the MCMC draws to a JMP
data file */
dt=Open( "$SAMPLE_DATA/bayesfitdf5.jmp" ); /* open the jmp data file as a visible data
table */
Column("mu")<<SetModelingType("Continuous"); /* for some reason columns come across as
ordinal */
Column("LNsigma")<<SetModelingType("Continuous");
dt<<New Column("sigma", Numeric, Continuous, Formula(exp(LNsigma)));
dt<<New Column("%CV", Numeric, Continuous, Formula(100*sigma/mu)); /* Add posterior of
CV */


/* The following produces a kernal density 95% bivariate Credible Interval for
mu,sigma*/
Bivariate(
       Y( :sigma ),
       X( :mu ),
       Show Points( 0 ),
       Nonpar Density(
              {Kernel Control( 1 ), Contour Fill( 1 ), Set Kernel( 2.1221, 0.13581 )}
       ),
       SendToReport(
```

```
		Dispatch( {}, "1", ScaleBox, {Show Major Grid( 1 ), Show Minor Grid( 1 )}
),
		Dispatch(
			{},
			"2",
			ScaleBox,
			{Min( -0.780131746717529 ), Max( 50.8159158517566 ), Minor Ticks(
3 ),
			Show Major Grid( 1 ), Show Minor Grid( 1 )}
		),
		Dispatch( {}, "Bivar Plot", FrameBox, {Frame Size( 332, 263 )} )
	)
);

/* The following gives a kernal density estimate for %CV plus a 95% credible interval
*/
Distribution(
	Continuous Distribution(
		Column( :Name( "%CV" ) ),
		Horizontal Layout( 1 ),
		Vertical( 0 ),
		Density Axis( 1 ),
		Outlier Box Plot( 0 ),
		Fit Distribution( Smooth Curve )
	),
	SendToReport(
		Dispatch(
			{"%CV"},
			"1",
			ScaleBox,
			{Max( 20.5 ), Inc( 1 ), Minor Ticks( 1 )}
		),
		Dispatch( {"%CV"}, "Distrib Histogram", FrameBox, {Frame Size( 378, 301
)} )
	)
);



/* The following plots the MCMC chain sequence and provides an autocorrelation
analysis*/
New Window( "bayesfitdf - Time Series",
	V List Box(
		Time Series( Y( :mu ) ),
		Time Series( Y( :sigma ) )
	)
);

/* The following gives kernal density estimates and credible intervals */
Distribution(
	Continuous Distribution( Column( :mu ), Fit Distribution( Smooth Curve ) ),
	Continuous Distribution( Column( :sigma),      Fit Distribution( Smooth Curve ))
);
```

Summary and Conclusions

The advantages of Bayesian approaches are becoming better known and it is likely that Bayesian analyses will join sampling theory methods in the mainstream of data analysis within the next decade. It can be expected that Bayesian concepts and methods will bring new insights, experimental efficiencies, and better risk assessment, modeling, and prediction paradigms to engineers, scientists and decision makers. So it is important for data analysts to understand and add these tools to their current “tool kit”. The JMP R interface provides a convenient mechanism for JMP users to become familiar with Bayesian approaches.

Some simple Bayesian analyses can be done directly in JMP, but more complex analyses require use of the R interface to execute MCMC. The freely available R and WinBUGS packages as well as some R libraries must be

present on the user's computer. Some familiarity with JSL as well as the R and WinBUGS languages is needed. This paper gives a short primer on these aspects as well as some examples to help JMP users move up the learning curve.

The new JMP R interface gives users freedom and access to powerful analytic tools. However with freedom comes responsibility. Use of Bayesian analysis, like any data analysis, demands attention to verification of details and assumptions. When using unfamiliar data analytic techniques, it is easy to make mistakes. So it is important to try known examples and verify good understanding of the concepts and potential pitfalls before using these methods for real problems. It is probably best to close this paper with a cautionary statement drawn largely from a famous quote in the WinBUGS manual that comes with the software:

> Potential users are reminded to be extremely careful when using Bayesian approaches for serious statistical analysis. Careful prior elicitation, verifying that the posterior is a proper distribution, appropriate parameterization, appropriate MCMC method choice, monitoring for convergence, and model checking are very important and are the user's responsibility. Be particularly careful with types of models for which you do not have a precedent example. If there is a problem, MCMC software might just crash, which is not very good, but it might well carry on and produce answers that are wrong, which is even worse. Beware - Gibbs sampling can be dangerous!

References

1. Andersen HC, in Wullschlager J (Ed.); Nunnally, T (Transl.) (2005). Fairy Tales. New York: Viking. ISBN 0-670-03377-4.
2. Dale AI (2003) Most honorable remembrance: The life and works of Thomas Bayes , Springer-Verlag, New York, ISBN 0-387-00499-8
3. Goodman SN. (1999) Toward evidence-based medical statistics. 1: The P value fallacy, and 2. The Bayes factor, Ann Intern Med. 1999;130 (12):995-1013.
4. Vangel MG (1996) Confidence intervals for a normal coefficient of variation, The American Statistician 15(1), 21-26.
5. ICH Q8(R2) Pharmaceutical Development, Q9 Quality Risk Management, and Q10 Pharmaceutical Quality System, available from http://www.ich.org/cache/compo/276-254-1.html.
6. FDA GMP for 21st century. Available from http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/Manufacturing/QuestionsandAnswersonCurrentGoodManufacturingPracticescGMPforDrugs/UCM176374.pdf
7. LeBlond, D (2008) Estimation: Knowledge Building with Probability Distributions, Journal of GXP Compliance, Vol. 12 (4), 42-59, 2008. See Journal of Validation Technology, Vol. 14, No. 5, 2008 for correction to Table IV.
8. Berger R L and Hsu J C (1996) Bioequivalence trials, intersection-union tests and equivalence confidence sets, Statistical Science 11(4) 283-319.
9. Couzin J (2004), The new math of clinical trials Science 303 (5659), 784 - 786
10. FDA (2010) Guidance for the Use of Bayesian Statistics in Medical Device Clinical Trials available from http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm071121.pdf
11. Bolstad W (2007) Introduction to Bayesian Statistics, 2nd edition, John Wiley and Sons, Inc, Hoboken, NJ
12. Gelman A, Carlin JB, Stern HS, Rubin DB () Bayesian Data Analysis, 2nd edition, Chapman and Hall/ CRC, Boca Raton, FL
13. Albert J (2008) Bayesian Computation in R, corrected 3rd printing, Springer
14. Box GEP and Tiao GC (1992) Bayesian inference in statistical analysis, Wiley classic library edition, John Wiley and Sons, Inc, NY

Acknowledgements

I would like to thank Mark Bailey and Jeff Perkinson of SAS Institute for introducing me to JMP's R interface and guiding me through my first use of the scripting language. I also am indebted to John Wass for his patience and for the opportunity to contribute this paper.

Contact Information

Your comments and questions are valued and encouraged. The user will be happy to provide the accompanying power point presentation when available. Contact the author at:

Dave LeBlond

3091 Midlane Drive
Wadsworth, IL 60083
Phone: 847-935-1899
E-mail: david.leblond@abbott.com or david.leblond@sbcglobal.net