Bayesian Methods with SAS®

Special Collection

Foreword by
Danny Modlin

The correct bibliographic citation for this manual is as follows: Modlin, Danny. 2022. *Bayesian Methods with SAS®: Special Collection*. Cary, NC: SAS Institute Inc.

Bayesian Methods with SAS®: Special Collection

SAS Institute Inc., SAS Campus Drive, Cary, NC 27513-2414

August 2022

Table of Contents

Free SAS® e-Books: Special Collection

In this series, we have carefully curated a collection of papers that introduces and provides context to the various areas of analytics. Topics covered illustrate the power of SAS solutions that are available as tools for data analysis, highlighting a variety of commonly used techniques.

Discover more free SAS e-books!
support.sas.com/freesasebooks

sas.com/books
for additional books and resources.

sas® THE POWER TO KNOW®

Foreword

Despite many believing that Bayesian analysis is a new statistical method, it has been around since the 18th century. It got its beginning from Rev. Thomas Bayes, a Presbyterian minister, who was interested in the reversal of conditional probabilities. As computers became stronger and computer simulation provided calculational assistance, Bayesian analysis has become more prevalent. The heart of this analysis is the ability to incorporate outside information, called prior information, into the analysis rather than just letting the data drive the work. This prior information can be gathered from subject experts or previously concluded experiments.

When using Bayesian analysis, you are not actually changing the type of problem you are solving. For example, if you are a user of linear regression, you are still performing linear regression in a Bayesian viewpoint. The difference is how you perceive the parameters in the problem. This change in perception allows the final solution for these parameters to be called posterior distributions, where you are allowed to speak probabilistically unlike in a classical analysis. This yields a way to answer different styles of questions from your analysis.

Thanks to advancements in computer technology, statisticians across all fields are now attempting to invoke Bayesian techniques within their analyses. As applying Bayesian techniques becomes easier, the benefits of the alternative viewpoint of the parameters have sparked renewed interest by analysts. Please do not that feel that you must fully diverge from classical analysis. Bayesian analysis is an additional statistical tool that many like having in their toolbox.

SAS offers many different solutions to estimate probability using Bayesian methods, and several groundbreaking papers have been written to demonstrate these techniques. We have carefully selected a handful of these from recent SAS Global Forum papers to introduce you to the topics and let you sample what each has to offer.

Bayesian Concepts: An Introduction

By John Amrhein and Fei Wang

You employ Bayesian concepts to navigate your everyday life, perhaps without being aware that you are doing so. You rely on past experiences to assess risk, assign probable cause, navigate uncertainty, and predict the future. Yet, as a statistician, economist, epidemiologist, or data scientist, you hold tight to your frequentist methods. Why? This paper explores the philosophy of Bayesian reasoning, explains advantages to applying Bayes' rule, and confronts the criticism of subjective Bayesian priors.

An Introduction to Bayesian Analysis with SAS/STAT® Software

by Maura Stokes, Fang Chen, and Funda Gunes

The use of Bayesian methods has become increasingly popular in modern statistical analysis, with applications in numerous scientific fields. In recent releases, SAS® has provided a wealth of tools for Bayesian analysis, with convenient access through several popular procedures as well as the MCMC procedure, which is designed for general Bayesian modeling. This paper introduces the principles of Bayesian inference and reviews the steps in a Bayesian analysis. It then describes the built-in Bayesian capabilities provided in SAS/STAT®, which became available for all platforms with SAS/STAT 9.3, with examples from the GENMOD and PHREG procedures. How to specify prior distributions, evaluate convergence diagnostics, and interpret the posterior summary statistics is discussed.

Introducing the BGLIMM Procedure for Bayesian Generalized Linear Mixed Models

by Amy Shi and Fang Chen

SAS/STAT® 15.1 includes PROC BGLIMM, a new, high-performance, sampling-based procedure that provides full Bayesian inference for generalized linear mixed models. PROC BGLIMM models data from exponential family distributions that have correlations or nonconstant variability; uses syntax similar to that of the MIXED and GLIMMIX procedures (the CLASS, MODEL, RANDOM, REPEATED, and ESTIMATE statements); deploys optimal sampling algorithms that are parallelized for performance; handles multilevel nested and non-nested random-effects models; and fits models to multivariate or longitudinal data that contain repeated measurements. PROC BGLIMM provides convenient access, with improved performance, to Bayesian analysis of complex mixed models that you could previously perform with the MCMC procedure. This paper describes how to use the BGLIMM procedure for estimation, inference, and prediction.

Bayesian Analysis of GLMMs Using PROC BGLIMM

by Walter Stroup

Over the past two decades, generalized linear models (GLMMs) mixed models for non-normal data such as proportions, counts, time to event, and so forth have become standard tools for statistical analysis. Since its introduction, PROC GLIMMIX has been the primary GLMM procedure for SAS/STAT. The most recent edition of *SAS® for Mixed Models* includes three chapters on using GLIMMIX for GLMMs. PROC GLIMMIX is an excellent frequentist tool. However, Bayesian approaches are becoming increasingly important. Many academic journals prefer some even require Bayesian analysis. Even when not required, Bayesian methods allow us to use what we know prior to, or in the early stages of, an investigation. PROC BGLIMM is a new SAS/STAT procedure that makes Bayesian implementation of GLMMs relatively easy. BGLIMM uses syntax similar to PROC GLIMMIX, but there are some differences. This tutorial presents what you need to know to get started using PROC BGLIMM. We will use GLMM examples from *SAS for Mixed Models*, but with a Bayesian twist.

Bayesian Networks for Causal Analysis

by Fei Wang and John Amrhein

Bayesian Networks (BN) are a type of graphical model that represent relationships between random variables. The networks can be very complex with many layers of interactions. Graphical models become BNs when the relationships are probabilistic and uni-directional. Building BNs for causal analyses is a natural and reliable way of expressing (and confirming or refuting) our belief and knowledge about cause and effects. In addition, BNs can be easily reconfigured with minor modifications to facilitate our understanding of probabilistic mechanisms. This paper describes the construction of BNs for causal analyses and how to infer causal structures from observational and interventional data. The paper includes applications of causal BNs for classification using the HPBN Classifier node in Enterprise Miner. Visualization, inferences, and scenario analyses for the examples are discussed.

The Bayesians are Coming! The Bayesians are Coming! The Bayesians are Coming to Time Series!

by Aric LaBarr

With the computational advances over the past few decades, Bayesian analysis approaches are starting to be fully appreciated. Forecasting and time series also have Bayesian approaches and techniques, but most people are unfamiliar with them due to the immense popularity of Exponential Smoothing and ARIMA classes of models. However, Bayesian modeling and time series analysis have a lot in common! Both are based on using historical information to help inform future modeling and decisions. This talk will compare the classical Exponential Smoothing and ARIMA class models to Bayesian models with autoregressive components. It will compare results from each of the classes of models on the same data set as well as discuss how to approach Bayesian time series models in SAS.

Incorporating Auxiliary Information into Your Model Using Bayesian Methods in SAS® Econometrics

by Matthew Simpson

In addition to data, analysts often have available useful auxiliary information about inputs into their model—for example, knowledge that high prices typically decrease demand or that sunny weather increases outdoor mall foot traffic. If used and incorporated correctly into the analysis, the auxiliary information can significantly improve the quality of the analysis. But this information is often ignored. Bayesian analysis provides a principled means of incorporating this information into the model through the prior distribution, but it does not provide a road map for translating auxiliary information into a useful prior. This paper reviews the basics of Bayesian analysis and provides a framework for turning auxiliary information into prior distributions for parameters in your model by using SAS® Econometrics software. It discusses common pitfalls and gives several examples of how to use the framework.

Please realize that this is not the end. As more statisticians bring Bayesian into their analyses, these topics will expand in depth and breadth. For more resources, including presentations and additional papers, please use this link to visit the Bayesian Analysis GitHub: https://github.com/statmike/Bayesian-Analysis--Primarily-SAS-.

We hope these selections give you a useful overview of the many tools and techniques that are available to incorporate Bayesian methods into your analysis.

Danny Modlin is a Senior Analytical Training Consultant at SAS world headquarters in Cary, North Carolina. Since starting at SAS in 2011, Danny has taught and developed courses that span across many areas of statistics and SAS platforms, with a specialization in the application of Bayesian analyses. Danny received his Bachelor of Science in Mathematics from Elon College (now Elon University), a Masters of Mathematics from the University of North Carolina at Wilmington, and a Masters of Statistics from North Carolina State University.

Prior to his time at SAS, Danny was a mathematics, statistics, and computer science teacher at the middle school, high school, and collegiate levels. Outside of SAS, Danny's interests include local sports and meteorology.

Paper 1863-2018

Bayesian Concepts: An Introduction

John Amrhein and Fei Wang, McDougall Scientific Ltd.

ABSTRACT

You employ Bayesian concepts to navigate your everyday life, perhaps without being aware that you are doing so. You rely on past experiences to assess risk, assign probable cause, navigate uncertainty, and predict the future. Yet, as a statistician, economist, epidemiologist, or data scientist, you hold tight to your frequentist methods. Why? This paper explores the philosophy of Bayesian reasoning, explains advantages to applying Bayes' rule, and confronts the criticism of subjective Bayesian priors.

INTRODUCTION

This paper is concerned with using statistics for decision support. Specifically, when one is faced with making a decision on a course of action, he/she assesses the probabilities of possible outcomes when there is uncertainty about those outcomes. In this context, statistics is reliant on probabilities. Within the profession of statistics, there are two schools of thought regarding the definition and interpretation of probabilities; frequentist, and Bayesian.

Bayesian reasoning combines past experience with current information to assign probable cause and assess risk of an (un)wanted effect. For example, you may know a street in your town where you can park your car without paying for a permit because past experience indicates you will not get caught. Elsewhere, you pay for the permit because the risk is too high that you will get caught and have to pay a fine. In either case you combine your experiential knowledge with the known state of current affairs to decide whether to pay for a temporary permit. We process information in a similar manner to conclude probable cause for an illness or injury by considering past experience and gathering information from subject matter experts and peers. Bayes' Theorem is a mathematical construct that reflects this manner of processing information and making decisions.

Bayesian statistical methods are often described by how they differ from frequentist methods. There are a few fundamental differences:

	Frequentist	**Bayesian**
Population parameters	Fixed but unknown	Random
Experimental data	Random yet repeatable	Fixed
Inferences	$P\{\mathbf{X} \mid H_0(\theta)\}$	$P\{\theta \mid \mathbf{X}\}$
Interpretation	Mathematical frequency: Over repeated sampling, an estimated interval will capture the mean, say, 95 out of 100 times.	Probabilistic belief: The probability that the parameter's value is within the interval is, say, 0.95.
Prior information	Ignored or used indirectly (e.g. estimating sample sizes)	Directly incorporated via $P\{\theta\}$

Frequentist methods might have their highest value in tightly controlled experimental situations (rolling dice), when repeatability is possible, even probable. However, in situations in which repeatability is questionable, for example in biological studies, Bayesian methods are advantageous because data are treated as fixed and repeatability is not required. On the other hand, the Bayesian interpretation of range estimates may be preferable in all situations.

EARLY HISTORY

The first half of the 18th century saw religion versus science and mathematics argued in pamphlets by such theologians, philosophers and mathematicians as George Berkeley and David Hume. Hume, especially, argued that cause and effect can only be learned through observation, and not through tradition, beliefs or reasoning and logic. Although the essence of Hume's argument is still debated today, there is agreement that he believed that empirical evidence is required of correlated events for cause and effect to be established. Hume's treatise precipitated discomfort among the faithful who considered God to be the First Cause; you cannot observe a deity raising the sun each morning.

The Presbyterian Reverend Thomas Bayes attempted to counter Hume's claims mathematically. Amidst this environment of controversy, and using the fledgling math of probabilities, Bayes determined to find cause from observed effects rather than concluding the effects from an assumed cause; for example, perhaps it is sufficient to repeatedly observe the rising sun to conclude First Cause. Bayes began with a simple Boolean system to determine the position of an original ball on a table by observing whether a new ball thrown onto the table lands left or right of the original, while remaining blinded to the original ball's position. His system, which is not unlike Boolean search algorithms, had characteristics that we rely on today for causal analyses:

- A prior belief, with no corroborating evidence (data): the assumed initial position of the original ball
- Increasing amounts of evidence (data): each new ball's position relative to the original ball
- Updated belief (estimate): updated original ball's position given relative position of the new ball
- A resulting probability statement: probable position (region) of the original ball

This simple experiment reversed probabilities: rather than estimating the probability that a new ball would land left or right of the original ball, given the original ball's position, Bayes estimated the probability of the original ball's position using increasing amounts of observed data.

Whether through modesty or dissatisfaction with, or uncertainty about, his own work, Bayes did not promote or publish his findings. About a decade later, another Presbyterian Minister, Richard Price, published a refined version of Bayes' work on inverse probabilities in an attempt to prove the existence of a wise deity (the cause) by observing natural laws (the effects).

Independent of Bayes' and Price's work, and later in the 18th century, Pierre Simon Laplace, after reading Abraham de Moivre's *Doctrine of Chances* (the same book studied by Bayes), adopted the mathematics of probabilities to deal with variation in astronomical data relating to the positions of the planets and the sun. Laplace called his method the "probability of cause." In 1814, after decades of working with data, he published *Essai philosophique sur les probabilités* in which he specified principles of probabilities, one of which we would recognize as Bayesian probability: the probability of a cause given an event is proportional to the probability of the event given the cause.

Sharon Bertsch McGrayne writes a captivating history of Bayes' Theorem, from Bayes to Laplace to the present day. It is clear that Laplace laid the foundation for Bayes' Theorem and its application. The theorem, however, bears the name of his predecessor, a moniker that was first applied during the 1950s.

SUBJECTIVITY AND PRIORS

There are several ways to write conditional probabilities and Bayes' Rule, each being useful in a different context. Assuming *A* and *B* are observable events, P() means probability, (*A*,*B*) means *A* and *B* occur together, and (*A*|*B*) means *A* given that *B* has been observed, then:

$$P(A|B) = \frac{P(A,B)}{P(B)} \quad \text{or} \quad P(A,B) = P(A|B)P(B) = P(B|A)P(A) \qquad (1)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \text{or} \quad P(A|B) = P(A)\frac{P(B|A)}{P(B)} \qquad (2)$$

$$P(A|B) = \frac{P(B|A)P(A)}{\sum P(B|A_i)P(A_i)} \quad \text{or} \quad P(A|B) = P(A)\frac{P(B|A)}{\sum P(B|A_i)P(A_i)} \qquad (3)$$

The algebraic inversion of probabilities in (1) gives rise to (2). Equation (3) stems from (2) by substituting the unconditional (marginal) probability of event *B* with the sum of its partitions over all possible outcomes of event *A*. Equations (2) and (3) are useful when we interpret Bayesian methods as updating prior probabilities with new evidence. An interpretation is:

> My updated (posterior) belief of event *A* given that I have observed event *B,* P(*A*|*B*), is dependent upon my prior belief of event *A,* P(*A*), and the likelihood of event *B* given that event *A* has occurred, P(*B*|*A*)/P(*B*).

For example, if event *A* represents a visit to a webpage with a SAS® advertisement, and *B* represents a purchase of SAS® software, then a reasonable question is whether the advertisement prompts viewers to purchase. The necessary pieces of information to estimate P(*visit* | *purchase*) include the probability that one visits the webpage, P(*visit*), and, for each purchase of software, whether the webpage was visited, P(*purchase* | *visit*) and P(*purchase* | *no visit*).

This example is one of cause and effect. In general, we are interested in P(cause | effect). In this example, the hypothesis of interest is that a visit to the webpage (and presumably seeing the advertisement), causes one to purchase the software; P(*visit* | *purchase*). Note that there is a temporal element in that the cause must precede the effect. The difficulty will be to assign the probability of visiting the webpage, P(*visit*), which is the prior probability to be updated with each purchase of software. How do you specify a prior, especially before any evidence or data are gathered; in this example, before anyone purchases the software?

When Laplace was made aware of Bayes' work (via Richard Price), he acknowledged Bayes' ingenuity in using an initial guess as to the position of the original ball on the table; i.e. before any observations were made. The subjectivity of the prior did not, presumably, cause Laplace any anxiety. Rather, he recognized that Bayes' initial estimate of the ball's position was needed to begin the learning algorithm of his theorem. Although the prior was subjective, the system learned with each iteration of observed data and corrected what may have been a poor estimate with a better estimate.

The subjective nature of priors has been a major complaint against Bayesian methods. However, frequentist methods are not devoid of subjectivity. During the planning of an experiment, effect sizes or variance estimates are needed to estimate sample size requirements. Inferences from hypothesis tests and confidence intervals rely on a pre-specified allowable type 1 error rate, which has been traditionally, and arbitrarily, set at 5%. In Bayesian analyses, you can use non-informative priors to mitigate the impact of subjectivity on the posterior estimates. Also, in operational systems in which priors are previous posteriors, subjectivity diminishes with time.

BELIEFS AND CREDIBILITY

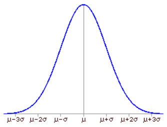
There are a number of benefits to adopting Bayesian methods:

- Formal, mathematical method to use prior information
- Data are not required to specify a prior; can rely on subject matter expertise
- Ability to make probability statements about probable causes
- Ability to conduct scenario analyses (simulations) in complex cause and effect systems
- Ease of interpretation

Frequentist interpretation of probabilities is a ratio of long-run frequencies. That is, the number of times an occurrence is observed given the number of trials. Bayesian interpretation of probabilities is a degree of belief. When you read the weather forecast, do you interpret "30% chance of rain" as a frequency? That is, during previous similar weather patterns, it rained 3 out of 10 times? Or do you interpret the probability as the degree to which, on a scale of 0 to 100, forecasters believe it will rain on this given day? The degree of belief is a more natural interpretation in non-repeatable situations.

Things become more abstract in a statistical setting. Suppose you are estimating a regression parameter. If you know the distribution of the parameter, given the data, then you can make probability, or degree of belief, statements about the parameter. This is in contrast to a range estimate, or confidence interval, with an associated statement about the “confidence” of long run frequencies. The difference between this two interpretations depends on what is considered to be random.

Modifying the table from this paper’s introduction:

$y = \alpha + \beta x$	**Frequentist**	**Bayesian**
Population parameter	β fixed but unknown	β random
Sample data	Random	Fixed
Inferences	P{**Y,X** | H_0(β=0)}	P{ β **| Y,X**}
	95% CI: (β_L , β_U) random about fixed β	95% Credible Region from distribution of β 
Interpretation	Over repeated sampling, an estimated interval will capture β 95% of the time.	The probability that β is between β_L and β_U is 0.95.

Here is an example of a quadratic regression. Published estimates were used to establish the prior distributions for a Bayesian analysis with new experimental data. The frequentist column of estimates ignores the prior published estimates and relies solely on the new data.

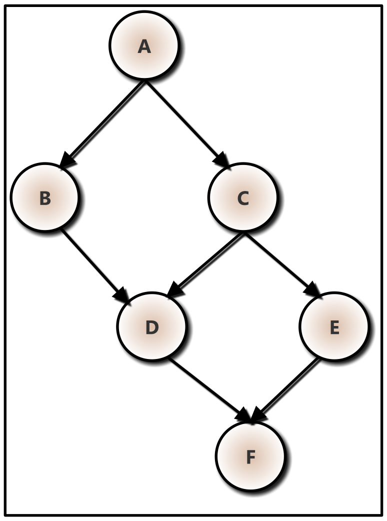
Parameter	Published estimates (priors)	Frequentist	Bayesian: Uniform Prior	Bayesian: Normal or Gamma Prior**
Intercept	24.60	22.54	22.51	23.24
Slope	6.06	7.56	7.57	7.33
Slope^2	-0.17	-0.28	-0.28	-0.27
Scale	10.90	12.91	13.10	12.50

** Normal used for coefficients, gamma used for the scale parameter

The 95% confidence interval for the slope parameter estimated using frequentist methods is 4.51 to 10.61. The interpretation of this interval is: “If I repeat this analysis with 100 independent samples, 95 of the estimated intervals for the slope parameter will include the true parameter value.”

The 95% Highest Posterior Density (HPD, see Bayesian Analyses and SAS below) for the slope parameter, using the informative priors, is 5.17 to 9.72. The interpretation for this interval is: “With 95% probability, the true value of the slope parameter is in this range.”

CAUSALITY AND BAYESIAN NETWORKS



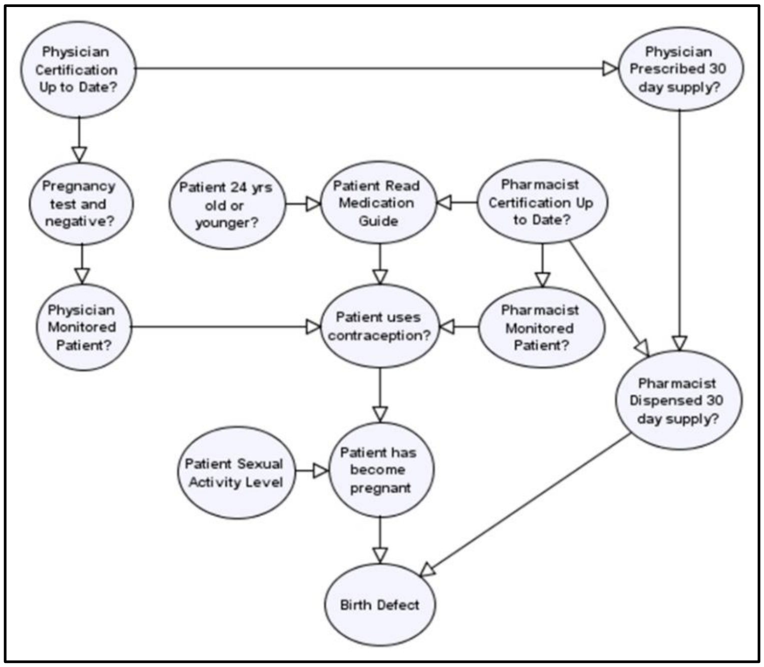
A type of model that is gaining prominence is the Bayesian Network (BN). BNs are Directed, Acyclic Graphs (DAG) made up of nodes, representing random variables, and arcs (edges, links, or connectors) representing probabilistic dependencies. They help define, via visualization, the dependencies in a complex system or network and are so named because of three attributes:

- The node distributions and relationships may be subjective
- Bayes' Theorem is repeatedly used to update node distributions
- Inferences are of a causal nature, rather than correlations

Bayesian probabilities are calculated for each node whenever new data become available, resulting in an updated probability of the occurrence of the outcome of interest, *F*, or a probable cause to *F* if it is known that *F* occurred. The interpretation of cause and effect derives from the single direction of the arcs (the "A" in DAG) and estimating the distributional dependencies of all random variables rather than just the target (as in frequentist regression methods). In the DAG on the left, the arcs leaving *A* are diverging connectors, so *A* is a common cause to *B* and *C*. The arcs entering *D* are converging connectors, so *D* is a common effect of *B* and *C*.

Conditional independence and the Markov Property (Compatibility) of BNs greatly increase the efficiency with which distributions are updated. Conditional independence states that if P(*A*,*C*) > 0 and P(*A*|*B*,*C*) = P(*A*|*C*), then events *A* and *B* are conditionally independent. That is, if it is possible that events *A* and *C* occur together, and knowing *B* does not provide any additional information to knowing *C* alone regarding the occurrence of *A*, then *A* and *B* are independent, conditional on knowing *C*. The Markov property states that, if a probability model correctly defines all of the conditional independencies represented by a BN, then that model and the BN are compatible. This is not unlike the assumption in regression that the specified model is correct.

Bayesian Networks have a natural application to risk management. They can incorporate subjective information from subject matter experts, assign probable cause in a scenario analysis or real occurrence of an undesirable event, and continually update the system as new information becomes available. Through the assignment of probable cause, BNs indicate how limited resources should be allocated to mitigate risks of the undesirable event.


Physician Certification Up to Date?
Physician Prescribed 30 day supply?
Pregnancy test and negative?
Patient 24 yrs old or younger?
Patient Read Medication Guide
Pharmacist Certification Up to Date?
Physician Monitored Patient?
Patient uses contraception?
Pharmacist Monitored Patient?
Pharmacist Dispensed 30 day supply?
Patient Sexual Activity Level
Patient has become pregnant
Birth Defect

The U.S. Food and Drug Administration (FDA) has the option to require pharmaceutical firms to plan and implement a Risk Evaluation and Mitigation Strategy (REMS) as a requirement for market approval for a new drug. FDA's draft guidance to industry states that *Part of risk management activities should include an objective, evidence-based evaluation of the efficacy in risk communication.* The BN on the left represents a strategy to prevent birth defects to children of female patients prescribed a new drug. The strategy of the fictitious company hinges on certifying physicians and pharmacists to prescribe and dispense the drug, their monitoring of patients, and a communication directly to the patient about the risks to unborn children. The effect of interest is a birth defect occurrence. The model includes possible causes as well as mitigation factors leading to the effect. The pharmaceutical firm can initiate the model using subject matter expertise (as did Bayes in 1763) and previously collected data. As the drug is prescribed and used in practice, the system updates factors' distributions to, in turn, update the probability of a birth defect. At any time, the pharmaceutical firm can analyze a scenario in which a birth defect occurred to assign probable cause. We can conduct a scenario analysis by setting the occurrence of a birth defect to "yes" and update all of the probability distributions. If, for example, the updated model indicates that the probability that the physician monitored the patient dropped by about 15%, which may have influenced the patient to relax her use of contraception, then to strengthen the risk mitigation program, more resources should be allocated to ensuring physicians monitor patients.

Financial institutions are also concerned with managing risks. The Basel Committee on Banking Supervision sets operational guidelines for financial institutions considered "too big to fail". Similar to FDA guidance, the Basel Committee publishes its recommendations and requirements. An example is *Principles for Effective Risk Data Aggregation and Risk Reporting*, in which Principle 7, in part, states:

> Approximations are an integral part of risk reporting and risk management. Results from models, scenario analyses, and stress testing are examples of approximations that provide critical information for managing risk. While the expectations for approximations may be different than for other types of risk reporting, banks should follow the reporting principles in this document and establish expectations for the reliability of approximations (accuracy, timeliness, etc) to ensure that management can rely with confidence on the information to make critical decisions about risk. This includes principles regarding data used to drive these approximations.

Bayesian Networks are a tool that banks can use to remain compliant with Basel principles and to effectively allocate resources to mitigate operational risks.

BAYESIAN ANALYSES AND SAS

From *Introduction to Bayesian Analysis Procedures* in the SAS/STAT® 14.3 User's Guide.

> SAS/STAT software provides Bayesian capabilities in six procedures: BCHOICE, FMM, GENMOD, LIFEREG, MCMC, and PHREG. The FMM, GENMOD, LIFEREG, and PHREG procedures provide Bayesian analysis in addition to the standard frequentist analyses they have always performed. …The BCHOICE procedure provides Bayesian analysis for discrete choice models. The MCMC procedure is a general procedure that fits Bayesian models with arbitrary priors and likelihood functions.

All applications of Bayesian methods require that you specify a prior. A few attributes of priors include the following:

- All priors are subjective in that they represent a degree of belief.
- Priors that have little impact on the posterior (flat compared to the likelihood) are known as *objective, flat, diffuse,* or *noninformative*
- Priors are probability functions and therefore should integrate to one. If they do not, they are known as *improper* priors (e.g. uniform over the real line). However, they are sometimes useful.
- Priors may lead to *improper posteriors*, which cannot be used for inference
- Priors which are flat within the appreciable range of the likelihood (e.g. ±2 or 3 sd of a normal mean) and have small values outside that range are said to be *locally uniform*. Jeffrey's prior is a useful prior in this category.
- *Conjugate* priors are priors that represent a family of distributions that lead to posteriors that are in the same family. That is, the posterior has the same distributional form as the prior. Examples include prior-likelihood combinations; Normal-normal, Beta-binomial, and Gamma-Poisson.

Inferences from Bayesian analyses rely on the posterior distribution of the parameter of interest. Point estimates can be the mean, median, or mode. Variances estimates arise from the posterior variance. Most problems cannot estimate the posterior distribution using a closed-form solution. Rather, simulation algorithms are used. All Bayes applications in SAS rely on Markov chain Monte Carlo (MCMC) methods. You need to be careful to not confuse the estimated standard error for the estimated mean of interest and the MCMC Standard Error (MCSE) due to the simulation. The standard error is, in part, determined by the sample size and the MCSE is a function of the number of simulation iterations.

There are two types of Bayesian intervals; 1) equal-tailed intervals, $100(\alpha/2)^{th}$ to $100(1-\alpha/2)^{th}$, of the posterior estimate, and 2) Highest Posterior Density Regions (HDR or HPD), which is the shortest possible interval on the parameter in which the density of any point within the interval is higher than of any point outside the interval. HPDs are often preferred because they tend to be shorter intervals than the equal-tailed intervals.

Hypothesis testing is also possible with Bayesian methods, but is not typically the objective.

BAYESIAN NETWORKS IN ENTERPRISE MINER

Bayesian Networks are available in SAS Enterprise Miner 14.1 via the HP Bayesian Network (HPBN) node (and in Factory Miner 14.1). A synthetic data set, “Asia”, from Lauritzen and Spiegelhalter (1988) illustrates results from the HPBN node.

> Shortness-of-breath (dyspnea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-ray do not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnea.

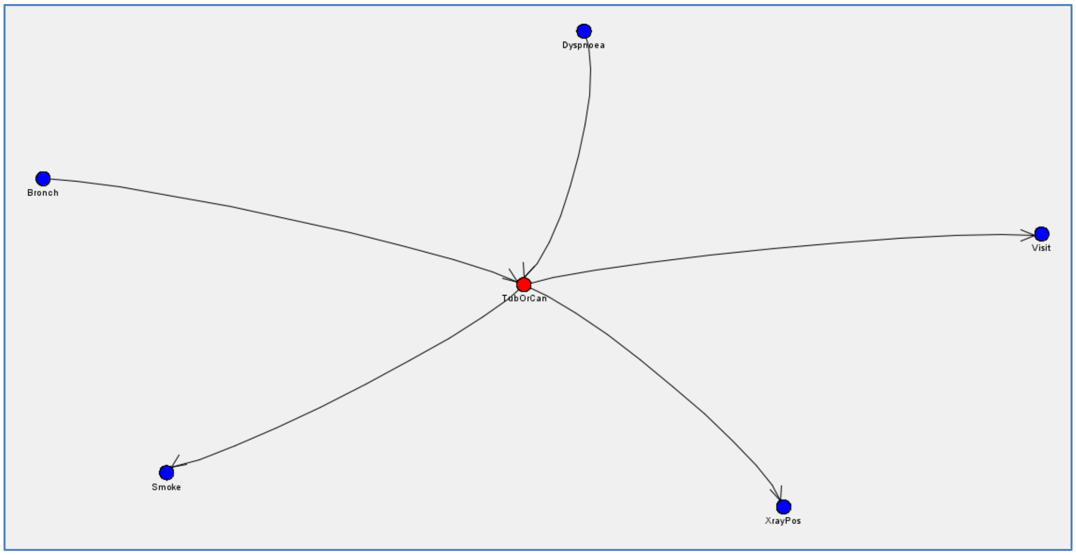
The data set contains 5000 observations and 8 binary character variables whose values are either yes or no. Suppose that we want to determine the dependencies between a set of inputs, dyspnea, bronchitis, a visit to Asia, smoking, and a positive x-ray, and the verified presence of either tuberculosis or lung cancer.

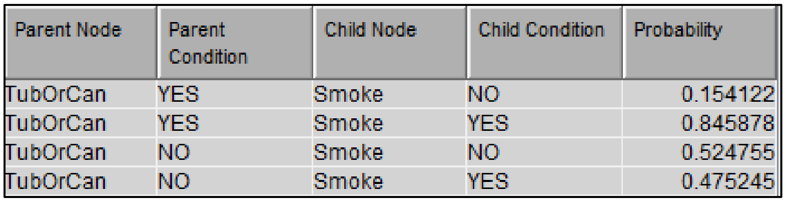
The Asia Data Set		
Variable	**Label**	**Description**
Dyspnea	Dyspnea	Presence of dyspnea?
Tuber	Tuberculosis	Presence of tuberculosis?
Cancer	Lung Cancer	Presence of lung cancer?
Bronch	Bronchitis	Presence of bronchitis?
Visit	Visit to Asia	Visit to Asia?
Smoke	Smoking	Smoker?
XrayPos	Chest X-ray	Positive chest X-ray?
TubOrCan	Tuberculosis or Lung Cancer	Presence of either tuberculosis or lung cancer?

We prepare and partition the data using other nodes preceding the HPBN node, including the metadata node to set the variable *TubOrCan* as the target. We configure the HPBN node to auto-select from four BN structure types and the possible models within each of those types. Possible models include not only whether a variable is in or out of the model, but the relationship between the nodes; be it a parent, child, or neighbor. HPBN fits the possible models using the Bayesian Information Criteria (BIC) and tests of independence and selects the best fitting model using the BIC on the validation data set.

For our implementation, we configure the HPBN node to not use the tuberculosis and cancer variables. We also turn off variable pre-screening and selection because we want to explore the dependencies among the other inputs. Therefore, our possible models are reduced to the various BN structure types and the edges connecting the nodes. More details are provided in a companion SASGF paper, 2776-2018, by Wang and Amrhein.

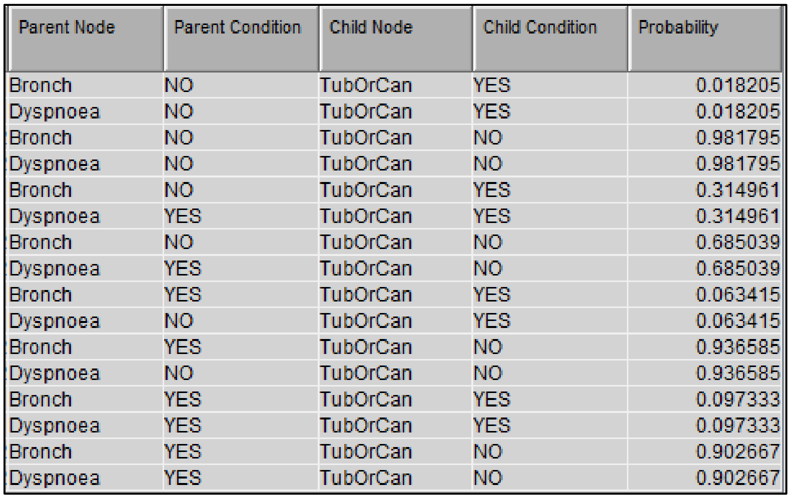
The final model is a Parent-Child Bayesian Network. The target variable is always at the center and is shown as red. The inputs are blue and the direction of the arcs are from parent to child. Posterior probabilities are given as P(child | parent). You must consider the temporal relationship, which HPBN does not know, to declare probable cause for an effect.





Parent Node	Parent Condition	Child Node	Child Condition	Probability
TubOrCan	YES	Smoke	NO	0.154122
TubOrCan	YES	Smoke	YES	0.845878
TubOrCan	NO	Smoke	NO	0.524755
TubOrCan	NO	Smoke	YES	0.475245

For example, the S*moke* node is a child of *TubOrCan*. The posterior conditional probability of smoking (cause) given tuberculosis or cancer (effect) is 0.846. Because we know that smoking pre-dates disease onset, **if this network completely represented all factors influencing disease onset**, then we could conclude, with very high probability, that smoking causes tuberculosis or cancer.



Parent Node	Parent Condition	Child Node	Child Condition	Probability
Bronch	NO	TubOrCan	YES	0.018205
Dyspnoea	NO	TubOrCan	YES	0.018205
Bronch	NO	TubOrCan	NO	0.981795
Dyspnoea	NO	TubOrCan	NO	0.981795
Bronch	NO	TubOrCan	YES	0.314961
Dyspnoea	YES	TubOrCan	YES	0.314961
Bronch	NO	TubOrCan	NO	0.685039
Dyspnoea	YES	TubOrCan	NO	0.685039
Bronch	YES	TubOrCan	YES	0.063415
Dyspnoea	NO	TubOrCan	YES	0.063415
Bronch	YES	TubOrCan	NO	0.936585
Dyspnoea	NO	TubOrCan	NO	0.936585
Bronch	YES	TubOrCan	YES	0.097333
Dyspnoea	YES	TubOrCan	YES	0.097333
Bronch	YES	TubOrCan	NO	0.902667
Dyspnoea	YES	TubOrCan	NO	0.902667

The two parents of *TubOrCan*, bronchitis and dyspnea, may be symptoms of tuberculosis or cancer. Therefore, the posterior conditional probabilities could be used as diagnostic aids. For example, from the table at the left, a patient who presents with shortness of breath but without bronchitis has a 0.315 probability of having tuberculosis or cancer. Stated another way, tuberculosis or cancer causes shortness of breath with probability 0.315.

CONCLUSION

Bayesian methods have suffered a lack of use for centuries due to at least two issues; 1) the subjectivity inherent in the methods and the interpretation of probabilities as "degrees of belief" led many to characterize them as unscientific, and 2) applying the methods was limited to problems for which the posterior distribution had a closed form solution. Advances in computer processing, search algorithms, and simulation methods has addressed the second issue, leading to a resurrection of a debate of the first issue. The many advantages discussed in this paper have greatly increased the research and application of Bayesian methods.

This paper discussed the philosophy and thought process motivating Bayesian methods. The benefits to adopting Bayesian analytical methods includes the abilities to:

- Interpret inferences in a manner more closely aligned with natural decision making
- Incorporate prior information and knowledge
- Incorporate subject matter expertise without corroborating data
- Assign probable causes, rather than just correlations, to effects
- Simulate scenarios in complex cause and effect systems

Disadvantages include the need to:

- Assign distributions to the prior information
- Execute complex computations involving simulations
- Recognize situations in which the posterior distribution is heavily influenced by a subjective prior

REFERENCES & RECOMMENDED READING

1. Basel Committee on Banking Supervision. (2013). *Principles for effective risk data aggregation and risk reporting.* Retrieved from http://www.bis.org/publ/bcbs239.pdf
2. Lauritzen, S. L. and D. J. Spiegelhalter (1988). *Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems.* Journal of the Royal Statistical Society. Series B (Methodological), Vol. 50, No. 2, pp. 157-224. Blackwell Publishing for the Royal Statistical Society
3. McGrayne, S. B. (2011). *The theory that would not die: How Bayes' rule cracked the Enigma code, hunted down Russian submarines, and emerged triumphant from two centuries of controversy.* New Haven: Yale University Press.
4. Pearl, J. (2009). *Causality: Models, reasoning, and inference* (2nd ed.). Cambridge: Cambridge University Press.
5. U.S. Department of Health and Human Services, Food and Drug Administration. (2009). *Format and content of proposed Risk Evaluation and Mitigation Strategies (REMS), REMS assessments, and proposed REMS modifications (Draft guidance for industry)*. Retrieved from http://www.fda.gov/downloads/Drugs/.../Guidances/UCM184128.pdf
6. Wang, Fei and John Amrhein (2018). *Bayesian Networks for Causal Analysis.* Proceedings of SAS Global Forum 2018, Denver Colorado, Paper 2776-2018. SAS Institute Inc., Cary, NC.

ADDITIONAL SUGGESTED READING

1. Bain, Robert. (2016). Are Our Brains Bayesian? *Significance*. 13:4, 14-19. The Royal Statistical Society. London.
2. Basel Committee on Banking Supervision. (2011). *Operational risk – Supervisory guidelines for the Advanced Measurement Approaches.* Retrieved from http://www.bis.org/publ/bcbs196.pdf
3. SAS Institute Inc. (2008). *SAS/STAT® 14.3 user's guide*. Cary, NC: SAS Institute Inc.
4. Savage, Leonard J. (1972). *The Foundations of Statistics*. (2nd revised ed.) New York: Dover Publications, Inc.

ACKNOWLEDGMENTS

Fei and I thank our fellow statisticians at McDougall Scientific Ltd. for their support and comments that improved this paper; Janet McDougall, Hong Chen, Jim Wang, and Blue Neustifter.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

John Amrhein
McDougall Scientific Ltd.
jamrhein@mcdougallscientific.com

Paper SAS400-2014

An Introduction to Bayesian Analysis with SAS/STAT® Software

Maura Stokes, Fang Chen, and Funda Gunes
SAS Institute Inc.

Abstract

The use of Bayesian methods has become increasingly popular in modern statistical analysis, with applications in numerous scientific fields. In recent releases, SAS® has provided a wealth of tools for Bayesian analysis, with convenient access through several popular procedures as well as the MCMC procedure, which is designed for general Bayesian modeling. This paper introduces the principles of Bayesian inference and reviews the steps in a Bayesian analysis. It then describes the built-in Bayesian capabilities provided in SAS/STAT®, which became available for all platforms with SAS/STAT 9.3, with examples from the GENMOD and PHREG procedures. How to specify prior distributions, evaluate convergence diagnostics, and interpret the posterior summary statistics is discussed.

Foundations

Bayesian methods have become a staple for the practicing statistician. SAS provides convenient tools for applying these methods, including built-in capabilities in the GENMOD, FMM, LIFEREG, and PHREG procedures (called the built-in Bayesian procedures), and a general Bayesian modeling tool in the MCMC procedure. In addition, SAS/STAT 13.1 introduced the BCHOICE procedure, which performs Bayesian choice modeling. With such convenient access, more statisticians are digging in to learn more about these methods.

The essence of Bayesian analysis is using probabilities that are conditional on data to express beliefs about unknown quantities. The Bayesian approach also incorporates past knowledge into the analysis, and so it can be viewed as the updating of prior beliefs with current data. Bayesian methods are derived from the application of Bayes' theorem, which was developed by Thomas Bayes in the 1700s as an outgrowth of his interest in inverse probabilities.

For events A and B, Bayes' theorem is expressed as

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$$

It can also be written as

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B|A)\Pr(A) + \Pr(B|\bar{A})\Pr(\bar{A})}$$

where $\bar{A}$ means *not* A. If you think of A as a parameter θ and B as data y, then you have

$$\Pr(\theta|y) = \frac{\Pr(y|\theta)\Pr(\theta)}{\Pr(y)} = \frac{\Pr(y|\theta)\Pr(\theta)}{\Pr(y|\theta)\Pr(\theta) + \Pr(y|\bar{\theta})\Pr(\bar{\theta})}$$

The quantity $\Pr(y)$ is the marginal probability, and it serves as a normalizing constant to ensure that the probabilities add up to unity. Because $\Pr(y)$ is a constant, you can ignore it and write

$$\Pr(\theta|y) \propto \Pr(y|\theta)\Pr(\theta)$$

Thus, the likelihood $\Pr(y|\theta)$ is being updated with the prior $\Pr(\theta)$ to form the posterior distribution $\Pr(\theta|y)$.

For a basic example of how you might update a set of beliefs with new data, consider a situation where researchers screen for vision problems in children in an after-school program. A study of 14 students chosen at random produces two students with vision issues. The likelihood is obtained from the binomial distribution:

$$L = \binom{14}{2} p^2 (1-p)^{12}$$

Suppose that the parameter p only takes values { 0.1, 0.12, 0.14, 0.16, 0.18, 0.20}. Researchers have prior beliefs about the probabilities of these values, and they assign them prior weights. Columns 1 and 2 in Table 1 contain the possible values for p and the prior probability weights, respectively. You can then compute the likelihoods for each of the values for p based on the study results, and then you can weight them with the corresponding prior weight. Column 5 contains the posterior values, which are the computed values displayed in column 4 divided by the normalizing constant 0.2501. Thus, the prior beliefs have been updated to a posterior distribution by accounting for the data obtained by the study. The posterior values are similar to, but different from, the likelihood.

Table 1 Empirical Posterior Distribution

p	Prior Weight	Likelihood	Prior x Likelihood	Posterior
0.10	0.10	0.257	0.0257	0.103
0.12	0.15	0.2827	0.0424	0.170
0.14	0.20	0.2920	0.0584	0.233
0.16	0.20	0.2875	0.0576	0.230
0.18	0.15	0.2725	0.0410	0.164
0.20	0.10	.2501	0.0250	0.100
Total	1		.2501	1

In a nutshell, this is what any Bayesian analysis does: it updates your beliefs about the parameters by accounting for additional data. You weight the likelihood for the data with the prior distribution to produce the posterior distribution. If you want to estimate a parameter θ from data $\mathbf{y} = \{y_1, \ldots, y_n\}$ by using a statistical model described by density $p(\mathbf{y}|\theta)$, Bayesian philosophy says that you can't determine θ exactly but you can describe the uncertainty by using probability statements and distributions. You formulate a prior distribution $\pi(\theta)$ to express your beliefs about θ. You then update those beliefs by combining the information from the prior distribution and the data, described with the statistical model $p(\theta|\mathbf{y})$, to generate the posterior distribution $p(\theta|\mathbf{y})$.

$$p(\theta|\mathbf{y}) = \frac{p(\theta, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{\int p(\mathbf{y}|\theta)\pi(\theta)d\theta}$$

The quantity $p(\mathbf{y})$ is the normalizing constant of the posterior distribution. It is also called the marginal distribution, and it is often ignored, as long as it is finite. Hence $p(\theta|\mathbf{y})$ is often written as

$$p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)\pi(\theta) = L(\theta)\pi(\theta)$$

where L is the likelihood and is defined as any function that is proportional to $p(\mathbf{y}|\theta)$. This expression makes it clear that you are effectively weighting your likelihood with the prior distribution. Depending on the influence of the prior, your previous beliefs can impact the generated posterior distribution either strongly (subjective prior) or minimally (objective or noninformative prior).

Consider the vision example again. Say you want to perform a Bayesian analysis where you assume a flat prior for p, or one that effectively will have no influence. A typical flat prior is the uniform,

$$\pi(p) = 1$$

and because the likelihood for the binomial distribution is written as

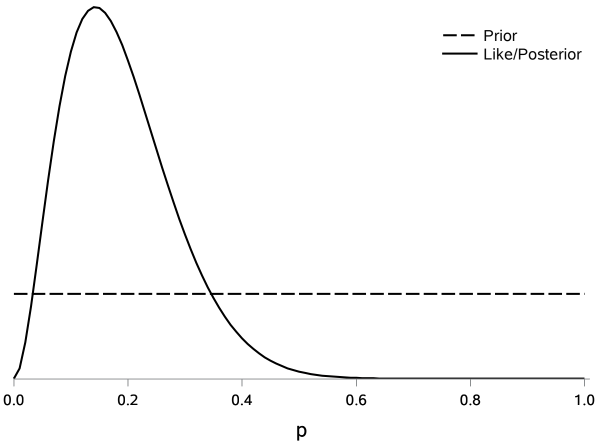
$$L(p) = \binom{n}{y} p^y (1-p)^{n-y}$$

you can write the posterior distribution as

$$\pi(p|y) \propto p^2(1-p)^{12}$$

which is also a beta (3,13) distribution. The flat prior weights equally on the likelihood, making the posterior distribution have the same functional form as the likelihood function. The difference is that, in the likelihood function, the random variable is y; in the posterior, the random variable is p. Figure 1 displays how the posterior distribution and the likelihood have the same form for a flat prior.

Figure 1 Beta (3,13) Posterior with Flat Uniform Prior



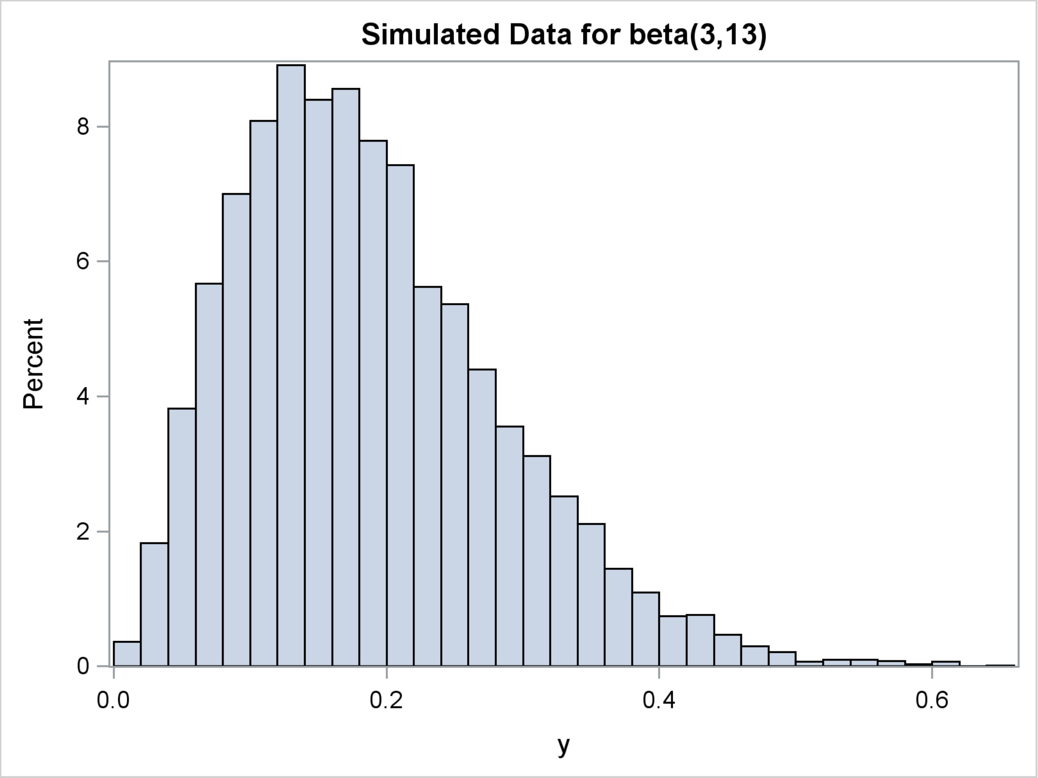
You can compute some summary measures of the posterior distribution directly such as an estimate of the mean of p and its variance, but you might want to compute other measures that aren't so straightforward, such as the probability that p is greater than a certain value such as 0.4. You can always simulate data from the beta distribution and address such questions by working directly with the simulated samples.

The following SAS statements create such a simulated data set for the beta (3,13) distribution:

```
data seebeta;
   %let N =10000;
   call streaminit (1234);
   a= 3; b= 13;
   do i=1 to &N;
     y = rand("beta", a, b );
   output;
   end;
run;
```

The results can be seen by the histogram in Figure 2 generated by using with the SGPLOT procedure.

Figure 2 Simulation for beta(3,13)



The mass of the distribution lies between 0.0 and 0.4, with the heaviest concentration between 0.1 and 0.2. Very little of the distribution lies beyond $p = 0.5$. If you want to determine the probability of $p > 0.4$, you would total the area under the curve for $p > 0.4$.

More often, closed forms for the posterior distribution like the beta distribution discussed above are not available, and you have to use simulation-based methods to estimate the posterior distribution itself, not just draw samples from it for convenience. Thus, the widespread use of Bayesian methods had to wait for the computing advances of the late 20th century. These methods involve repeatedly drawing samples from a target distribution and using the resulting samples to empirically approximate the posterior distribution. Markov chain Monte Carlo (MCMC) methods are used extensively. A Markov chain is a stochastic process that generates conditional independent samples according to a target distribution; Monte Carlo is a numerical integration technique that finds an expectation of an integral. Put together, MCMC methods generate a sequence of dependent samples from the target posterior distribution and compute posterior quantities of interest by using Monte Carlo. Popular and flexible MCMC simulation tools are the Metropolis, Metropolis-Hastings, and Gibbs sampling algorithms as well as numerous variations.

This paper does not discuss the details of these computational methods, but you can find a summary in the "Introduction to Bayesian Analysis" chapter in the *SAS/STAT User's Guide* as well as many references. However, understanding the need to check for the convergence of the Markov chains is essential in performing Bayesian analysis, and this is discussed later.

The Bayesian Method

Bayesian analysis is all about the posterior distribution. Parameters are random quantities that have distributions, as opposed to the fixed model parameters of classical statistics. All of the statistical inferences of a Bayesian analysis come from summary measures of the posterior distribution, such as point and interval estimates. For example, the mean or median of a posterior distribution provides point estimates for θ, whereas its quantiles provide *credible intervals.*

These credible intervals, also known as credible sets, are analogous to the confidence intervals in frequentist analysis. There are two types of credible intervals: the equal-tail credible interval describes the region between the cut-points for the equal tails that has $100(1-\alpha)\%$ mass, while the highest posterior density (HPD), is the region where the posterior probability of the region is $100(1-\alpha)\%$ and the minimum density of any point in that region is equal to or larger than the density of any point outside that region. Some statisticians prefer the equal-tail interval because it is invariant under transformations. Other statisticians prefer the HPD interval because it is the smallest interval, and it is more frequently used.

The prior distribution is a mechanism that enables the statistician to incorporate known information into the analysis and to combine that information with that provided by the observed data. For example, you might have expert opinion or historical information from previous studies. You might know the range of values for a particular parameter for biological reasons. Clearly, the chosen prior distribution can have a tremendous impact on the results of an analysis, and it must be chosen wisely. The necessity of choosing priors, and its inherent subjectivity, is the basis for some criticism of Bayesian methods.

The Bayesian approach, with its emphasis on probabilities, does provide a more intuitive framework for explaining the results of an analysis. For example, you can make direct probability statements about parameters, such as that a particular credible interval contains a parameter with measurable probability. Compare this to the confidence interval and its interpretation that, in the long run, a certain percentage of the realized confidence intervals will cover the true parameter. Many non-statisticians wrongly assume the Bayesian credible interval interpretation for a confidence interval interpretation.

The Bayesian approach also provides a way to build models and perform estimation and inference for complicated problems where using frequentist methods is cumbersome and sometimes not obvious. Hierarchical models and missing data problems are two cases that lend themselves to Bayesian solutions nicely. Although this paper is concerned with less sophisticated analyses in which the driving force is the desire for the Bayesian framework, it's important to note that the consummate value of the Bayesian method might be to provide statistical inference for problems that couldn't be handled without it.

Prior Distributions

Some practitioners want to benefit from the Bayesian framework with as limited an influence from the prior distribution as possible: this can be accomplished by choosing priors that have a minimal impact on the posterior distribution. Such priors are called noninformative priors, and they are popular for some applications, although they are not always easy to construct. An informative prior dominates the likelihood, and thus it has a discernible impact on the posterior distribution.

A prior distribution is noninformative if it is "flat" relative to the posterior distribution, as demonstrated in Figure 1. However, while a noninformative prior can appear to be more objective, it's important to realize that there is some degree of subjectivity in any prior chosen; it does not represent complete ignorance about the parameter in question. Also, using noninformative priors can lead to what is known as improper posteriors (nonintegrable posterior density), with which you cannot make inferences. Noninformative priors might also be noninvariant, which means that they could be noninformative in one parameterization but not noninformative if a transformation is applied.

On the other hand, an improper prior distribution, such as the uniform prior distribution on the number line, can be appropriate. Improper prior distributions are frequently used in Bayesian analysis because they yield noninformative priors and proper posterior distributions. To form a proper posterior distribution, the normalizing constant has to be finite for all $\mathbf{y}$.

Some of the priors available with the built-in Bayesian procedures are improper, but they all produce proper posterior distributions. However, the MCMC procedure enables you to construct whatever prior distribution you can program, and so you yourself have to ensure that the resulting posterior distribution is proper.

More about Priors

Jeffreys' prior (Jeffreys 1961) is a useful prior because it doesn't change much over the region in which the likelihood is significant and doesn't have large values outside that range—the local uniformity property. It is based on the observed Fisher information matrix. Because it is locally uniform, it is a noninformative prior. Thus, it provides an automated way of finding a noninformative prior for any parametric model; it is also invariant with respect to one-to-one transformations. The GENMOD procedure computes Jeffreys' prior for any generalized linear model, and you can use it for your prior density for any of the coefficient parameters. Jeffreys' prior can lead to improper posteriors, but not in the case of the PROC GENMOD usage.

You can show that Jeffreys' prior is

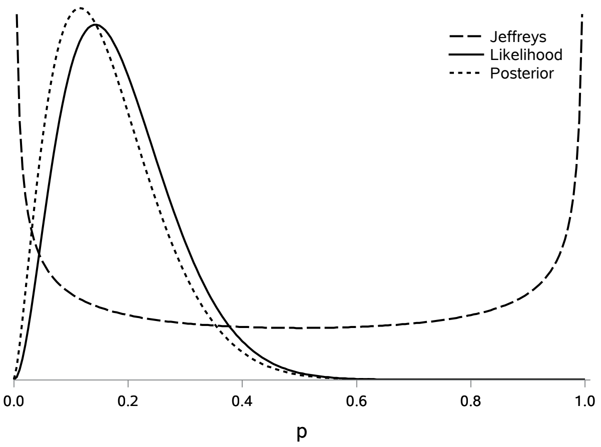
$$\pi(p) \propto p^{-1/2}(1-p)^{-1/2}$$

for the binomial distribution, and the posterior distribution for the vision example with Jeffreys' prior is

$$\begin{aligned} L(p)\pi(p) &\propto p^{y-\frac{1}{2}}(1-p)^{n-y-\frac{1}{2}} \\ &\sim \text{beta}(2.5, 12.5) \end{aligned}$$

Figure 3 displays how the Jeffreys' prior for the vision study example is relatively uninfluential in the areas where the posterior has the most mass.

Figure 3 Beta (2.5,12.5) Posterior with Jeffreys' Prior



A prior is a conjugate prior for a family of distributions if the prior and posterior distributions are from the same family. Conjugate priors result in closed-form solutions for the posterior distribution, enabling either direct inference or the construction of efficient Markov chain Monte Carlo sampling algorithms. Thus, the development of these priors was driven by the early need to minimize computational requirements. Although the computational barrier is no longer an issue, conjugate priors can still have performance benefits, and they are frequently used in Markov chain simulation because they directly sample from the target conditional distribution. For example, the GENMOD procedure uses conjugacy sampling wherever it is possible.

The beta is a conjugate prior for the binomial distribution. If the likelihood is based on the binomial (n, p):

$$L(p) \propto p^{y}(1-p)^{n-y}$$

and the prior is a beta (α, β),

$$\pi(p|\alpha,\beta) \propto p^{\alpha-1}(1-p)^{\beta-1}$$

then the posterior distribution is written as

$$\begin{aligned} \pi(p|\alpha,\beta,y,n) &\propto p^{y+\alpha-1}(1-p)^{n-y+\beta-1} \\ &= \text{beta}\,(y+\alpha, n-y+\beta) \end{aligned}$$

This posterior is easily calculated, and you can rely on simulations from it to produce the measures of interest, as demonstrated above.

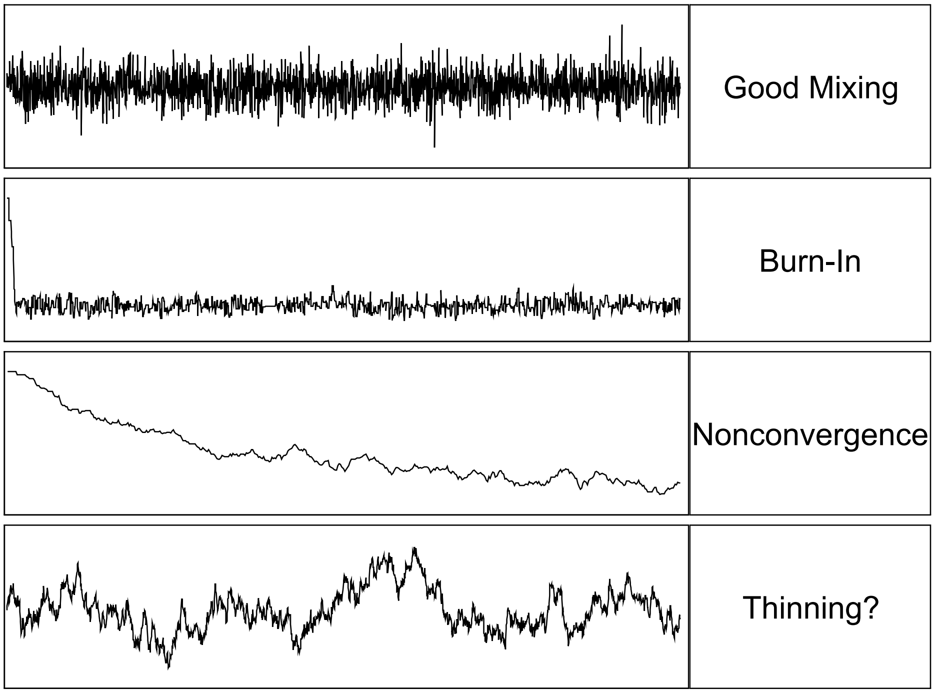
Assessing Convergence

Although this paper does not describe the underlying MCMC computations, and you can perform Bayesian analysis without knowing the specifics of those computations, it is important to understand that a Markov chain is being generated (its stationary distribution is the desired posterior distribution) and that you must check its convergence before you can work with the resulting posterior statistics. An unconverged Markov chain does not explore the parameter space sufficiently, and the samples cannot approximate the target distribution well. Inference should not be based on unconverged Markov chains, or misleading results can occur. And you need to check the convergence of all the parameters, not just the ones of interest.

There is no definitive way of determining that you have convergence, but there are a number of diagnostic tools that tell you if the chain hasn't converged. The built-in Bayesian procedures provide a number of convergence diagnostic tests and tools, such as Gelman-Rubin, Geweke, Heidelberger-Welch, and Raftery-Lewis tests. Autocorrelation measures the dependency among the Markov chain samples, and high correlations can indicate poor mixing. The Geweke statistic compares means from early and late parts of the Markov chain to see whether they have converged. The effective sample size (ESS) is particularly useful as it provides a numerical indication of mixing status. The closer ESS is to n, the better the mixing in the Markov chain. In general, an ESS of approximately 1,000 is adequate for estimating the posterior density. You might want it larger if you are estimating tail percentiles.

One of the ways that you can assess convergence is with visual examination of the trace plot, which is a plot of the sampled values of a parameter versus the sample number. Figure 4 displays some types of trace plots that can result:

Figure 4 Types of Trace Plots



By default, the built-in Bayesian procedures discard the first 2,000 samples as burn-in and keep the next 10,000. You want to discard the early samples to reduce the potential bias they might have on the estimates.(You can increase the number of samples when needed.) The first plot shows good mixing. The samples stay close to the high-density region of the target distribution; they move to the tail areas but quickly return to the high-density region. The second plot shows evidence that a longer burn-in period is required. The third plot sets off warning signals. You could try increasing the number of samples, but sometimes a chain is simply not going to converge. Additional adjustment might be required such as model reparameterization or using a different sampling algorithm. Some practitioners might see thinning, or the practice of keeping every kth iteration to reduce autocorrelation, as indicated. However, current practice tends to downplay the usefulness of thinning in favor of keeping all the samples. The Bayesian procedures produce trace plots and also autocorrelation plots and density plots to aid in convergence assessment.

For further information about these measures, see the “Introduction to Bayesian Analysis” chapter in the *SAS/STAT User’s Guide*.

Summary of the Steps in a Bayesian Analysis

You perform the following steps in a Bayesian analysis. Choosing a prior, checking convergence, and evaluating the sensitivity of your results to your prior might be new steps for many data analysts, but they are important ones.

1. Select a model (likelihood) and corresponding priors. If you have information about the parameters, use them to construct the priors.
2. Obtain estimates of the posterior distribution. You might want to start with a short Markov chain.
3. Carry out convergence assessment by using the trace plots and convergence tests. You usually iterate between this step and step 2 until you have convergence.
4. Check for the fit of the model and evaluate the sensitivity of your results due to the priors used.
5. Interpret the results: Do the posterior mean estimates make sense? How about the credible intervals?
6. Carry out further analysis: compare different models, or estimate various quantities of interest, such as functions of the parameters.

Bayesian Capabilities in SAS/STAT Software

SAS provides two avenues for Bayesian analysis: built-in Bayesian analysis in certain modeling procedures and the MCMC procedure for general-purpose modeling. The built-in Bayesian procedures are ideal for data analysts beginning to use Bayesian methods, and they suffice for many analysis objectives. Simply adding the BAYES statement generates Bayesian analyses without the need to program priors and likelihoods for the GENMOD, PHREG, LIFEREG, and FMM procedures. Thus, you can obtain Bayesian results for the following:

- linear regression
- Poisson regression
- logistic regression
- loglinear models
- accelerated failure time models
- Cox proportional models
- piecewise exponential models
- frailty models
- finite mixture models

The built-in Bayesian procedures apply the appropriate Markov chain Monte Carlo sampling technique. The Gamerman algorithm is the default sampling method for generalized linear models fit with the GENMOD procedure, and Gibbs sampling with adaptive rejection sampling (ARS) is generally the default, otherwise. However, conjugate sampling is available for a few cases, and the independent Metropolis algorithm and the random walk Metropolis algorithm are also available when appropriate.

The built-in Bayesian procedures provide default prior distributions depending on what models are specified. You can choose from other available priors by using the CPRIOR= option (for coefficient parameters) and SCALEPRIOR= option (for scale parameters). Other options allow you to choose the numbers of burn-ins, the number of iterations, and so on. The following posterior statistics are produced:

- point estimates: mean, standard deviation, percentiles
- interval estimates: equal-tail and highest posterior density (HPD) intervals
- posterior correlation matrix
- deviance information criteria (DIC)

All these procedures produce convergence diagnostic plots and statistics, and they are the same diagnostics that the MCMC procedure produces. You can also output posterior samples to a data set for further analysis. The following sections describe how to use the built-in Bayesian procedures to perform Bayesian analyses.

Linear Regression

Consider a study of 54 patients who undergo a certain type of liver operation in a surgical unit (Neter el al 1996). Researchers are interested in whether blood clotting score has a positive effect on survival.

The following statements create SAS data set SURGERY. The variable Y is the survival time, and LOGX1 is the natural logarithm of the blood clotting score.

```
data surgery;
   input x1 logy;
   y = 10**logy;
   label x1 = 'Blood Clotting Score';
   label y = 'Survival Time';
   logx1 = log(x1);
datalines;
6.7   2.3010
5.1   2.0043
..
..
;
run;
```

Suppose you want to perform a Bayesian analysis for the following regression model for the survival times, where ϵ is a $N(0, \sigma^2)$ error term:

$$Y = \beta_0 + \beta_1 logX_1 + \epsilon$$

If you wanted a frequentist analysis, you could fit this model by using the REG procedure. But this model is also a generalized linear model (GLM) with a normal distribution and the identity link function, so it can be fit with the GENMOD procedure, which offers Bayesian analysis. To review, a GLM relates a mean response to a vector of explanatory variables through a monotone link function where the likelihood function belongs to the exponential family. The link function g describes how the expected value of the response variable is related to the linear predictor,

$$g(E(y_i)) = g(\mu_i) = x_i^t \beta$$

where y_i is a response variable $(i = 1, \ldots, n)$, g is a link function, $\mu_i = E(y_i)$, x_i is a vector of independent variables, and β is a vector of regression coefficients to be estimated. For example, when you assume a normal distribution for the response variable, you specify an identity link function $g(\mu) = \mu$. For Poisson regression you specify a log link function $g(\mu) = \log(\mu)$, and for a logistic regression you specify a logit link function $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$.

The BAYES statement produces Bayesian analyses with the GENMOD procedure for most of the models it fits; currently this does not include models for which the multinomial distribution is used. The first step of a Bayesian analysis is specifying the prior distribution, and Table 2 describes priors that are supported by the GENMOD procedure.

Table 2 Prior Distributions Provided by the GENMOD Procedure

Parameter	Prior
Regression coefficients	Jeffreys', normal, uniform
Dispersion	Gamma, inverse gamma, improper
Scale and precision	Gamma, improper

You would specify a prior for one of the dispersion, scale, or precision parameters, in models that have such parameters.

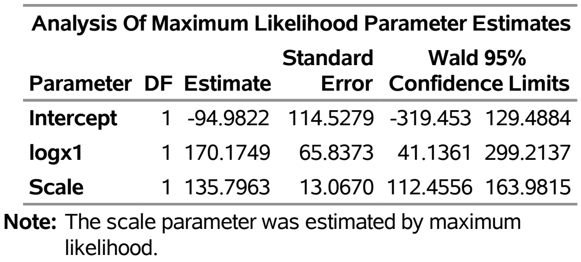
The following statements request a Bayesian analysis for the linear regression model with PROC GENMOD:

```
proc genmod data=surg;
   model y=logx1/dist=normal;
   bayes seed=1234 outpost=Post;
run;
```

The MODEL statement with the DIST=NORMAL option describes the simple linear regression model (the default is the identity link function.) The BAYES statement requests the Bayesian analysis. The SEED= option in the BAYES statement sets the random number seed so you can reproduce the analysis in the future. The OUTPOST= option saves the generated posterior samples to the POST data set for further analysis.

By default, PROC GENMOD produces the maximum likelihood estimates of the model parameters, as displayed in Figure 5.

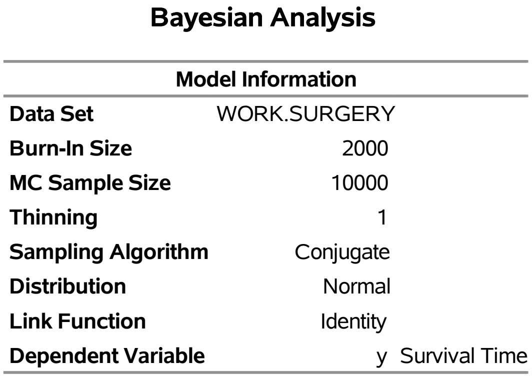
Figure 5 Maximum Likelihood Parameter Estimates



```
     Analysis Of Maximum Likelihood Parameter Estimates

                            Standard        Wald 95%
Parameter  DF  Estimate       Error   Confidence Limits
Intercept   1  -94.9822    114.5279   -319.453  129.4884
logx1       1  170.1749     65.8373    41.1361  299.2137
Scale       1  135.7963     13.0670   112.4556  163.9815

Note: The scale parameter was estimated by maximum
      likelihood.
```

Subsequent tables are produced by the Bayesian analysis. The "Model Information" table in Figure 6 summarizes information about the model that you fit. The 2,000 burn-in samples are followed by 10,000 samples. Because the normal distribution was specified in the MODEL statement, PROC GENMOD exploited the conjugacy and sampled directly from the target distribution.

Figure 6 Model Information



Bayesian Analysis

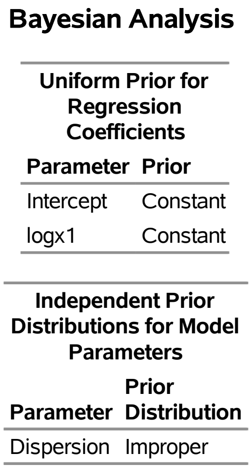
Model Information	
Data Set	WORK.SURGERY
Burn-In Size	2000
MC Sample Size	10000
Thinning	1
Sampling Algorithm	Conjugate
Distribution	Normal
Link Function	Identity
Dependent Variable	y Survival Time

The “Prior Distributions” table in Figure 7 identifies the prior distributions. The default uniform prior distribution is assumed for the regression coefficients β_0 and β_1, and the default improper prior is used for the dispersion parameter. An improper prior is defined as

$$\pi(\theta) \propto \frac{1}{\theta}$$

Both of these priors are noninformative.

Figure 7 Prior Distributions



Bayesian Analysis

Uniform Prior for Regression Coefficients

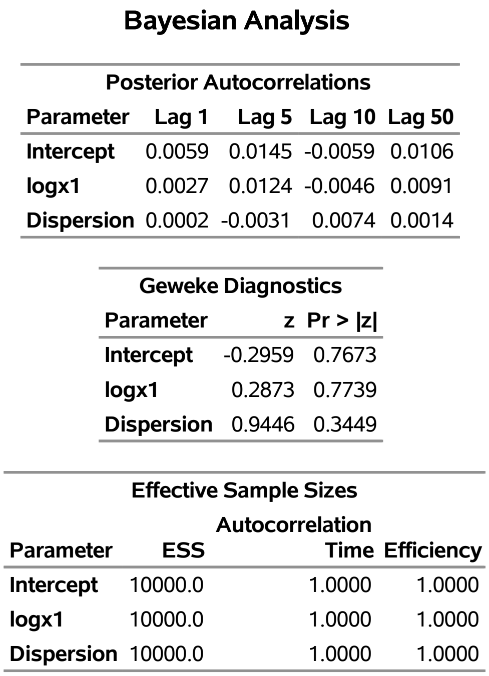
Parameter	Prior
Intercept	Constant
logx1	Constant

Independent Prior Distributions for Model Parameters

Parameter	Prior Distribution
Dispersion	Improper

Figure 8 displays the convergence diagnostics: the “Posterior Autocorrelations” table reports that the autocorrelations at the selected lags (1, 5, 10, and 50, by default) drop off quickly, indicating reasonable mixing of the Markov chain. The p-values in the “Geweke Diagnostics” table show that the mean estimate of the Markov chain is stable over time. The “Effective Sample Sizes” table reports that the number of effective sample sizes of the parameters is equal to the Markov chain sample size, which is as good as that measure gets.

Figure 8 Convergence Diagnostics Table



Bayesian Analysis

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Intercept	0.0059	0.0145	-0.0059	0.0106
logx1	0.0027	0.0124	-0.0046	0.0091
Dispersion	0.0002	-0.0031	0.0074	0.0014

Geweke Diagnostics

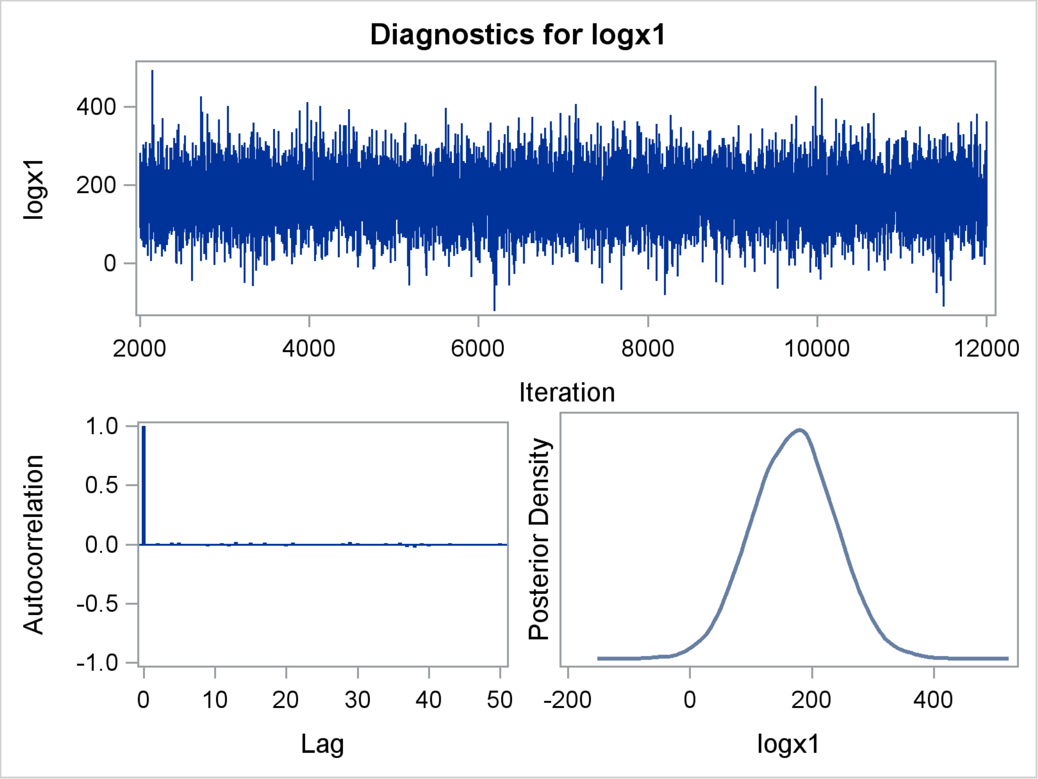
Parameter	z	Pr > |z|
Intercept	-0.2959	0.7673
logx1	0.2873	0.7739
Dispersion	0.9446	0.3449

Effective Sample Sizes

Parameter	ESS	Autocorrelation Time	Efficiency
Intercept	10000.0	1.0000	1.0000
logx1	10000.0	1.0000	1.0000
Dispersion	10000.0	1.0000	1.0000

The built-in Bayesian procedures produce three types of plots that help you visualize the posterior samples of each parameter. These diagnostics plots for the slope coefficient β_1 are shown in Figure 9. The trace plot indicates that the Markov chain has stabilized with good mixing. The autocorrelation plot confirms the tabular information, and the kernel density plot estimates the posterior marginal distribution. The diagnostic plots for the other parameters (not shown here) have similar outcomes.

Figure 9 Bayesian Model Diagnostic Plot for β_1



Because convergence doesn't seem to be an issue, you can review the posterior statistics displayed in Figure 10 and Figure 11.

Figure 10 Posterior Summary Statistics

Bayesian Analysis

Posterior Summaries

				Percentiles		
Parameter	**N**	**Mean**	**Standard Deviation**	**25%**	**50%**	**75%**
Intercept	10000	-95.9018	119.4	-176.1	-96.0173	-16.2076
logx1	10000	170.7	68.6094	124.2	171.1	216.7
Dispersion	10000	19919.9	4072.4	17006.0	19421.7	22253.7

Figure 11 Posterior Interval Statistics

Posterior Intervals

Parameter	**Alpha**	**Equal-Tail Interval**		**HPD Interval**	
Intercept	0.050	-328.7	135.5	-324.0	139.2
logx1	0.050	37.4137	304.3	36.8799	303.0
Dispersion	0.050	13475.4	29325.9	12598.8	28090.3

The posterior summaries displayed in Figure 10 are similar to the maximum likelihood estimates shown in Figure 5. This is because noninformative priors were used and the posterior is effectively the likelihood. Figure 11 displays the HPD interval and the equal-tail interval.

You might be interested in whether LOGX1 has a positive effect on survival time. You can address this question by using the posterior samples that are saved to the POST data set. This means that you can determine the conditional probability $\Pr(\beta_1 > 0 \mid \mathbf{y})$ directly from the posterior sample. All you have to do is to determine the proportions of samples where β_1 > 0,

$$\Pr(\beta_1 > 0 \mid \mathbf{y}) = \frac{1}{N} \sum_{t=1}^{N} I(\beta_1^t > 0)$$

where N = 10000 is the number of samples after burn-in and I is an indicator function, where $I(\beta_1^t > 0) = 1$ if $\beta_1^t > 0$, and 0 otherwise. The following SAS statements produce the posterior samples of the indicator function I by using the posterior samples of β_1 saved in the output data set POST:

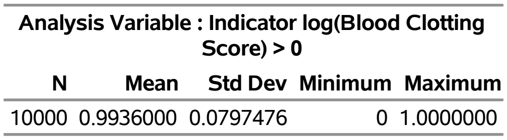
```
data Prob;
   set Post;
   Indicator = (logX1 > 0);
   label Indicator= 'log(Blood Clotting Score) > 0';
run;
```

The following statements request the summary statistics by using the MEANS procedure:

```
proc means data = prob(keep=Indicator);
run;
```

Figure 12 displays the results. The posterior probability that β_1 greater than 0 is estimated as 0.9936. Obviously LOGX1 has a strongly positive effect on survival time.

Figure 12 Posterior Summary Statistics with the MEANS Procedure



Analysis Variable : Indicator log(Blood Clotting Score) > 0				
N	Mean	Std Dev	Minimum	Maximum
10000	0.9936000	0.0797476	0	1.0000000

Logistic Regression

Consider a study of the analgesic effects of treatments on elderly patients with neuralgia. A test treatment and a placebo are compared, and the response is whether the patient reports pain. Explanatory variables include the age and gender of the 60 patients and the duration of the complaint before the treatment began.

The following SAS statements input the data:

```
data Neuralgia;
   input Treatment $ Sex $ Age Duration Pain $ @@;
datalines;
P  F  68   1  No   B  M  74  16  No  P  F  67  30  No
P  M  66  26  Yes  B  F  67  28  No  B  F  77  16  No
A  F  71  12  No   B  F  72  50  No  B  F  76   9  Yes
A  M  71  17  Yes  A  F  63  27  No  A  F  69  18  Yes
..
..
;
```

Logistic regression is considered for this data set:

$$\begin{aligned}
\text{pain}_i &\sim \text{binary}(p_i) \\
p_i &= \text{logit}(\beta_0 + \beta_1 \cdot \text{Sex}_{F,i} + \beta_2 \cdot \text{Treatment}_{A,i} \\
&\quad + \beta_3 \cdot \text{Treatment}_{B,i} + \beta_4 \cdot \text{Sex}_{F,i} \cdot \text{Treatment}_{A,i} \\
&\quad + \beta_5 \cdot \text{Sex}_{F,i} \cdot \text{Treatment}_{B,i} + \beta_6 \cdot \text{Age} + \beta_7 \cdot \text{Duration})
\end{aligned}$$

where Sex_F, Treatment_A, and Treatment_B are dummy variables for the categorical predictors.

You might consider a normal prior with large variance as a noninformative prior distribution on all the regression coefficients:

$$\pi(\beta_0, \cdots, \beta_7) \sim \text{normal}(0, \text{var} = 1e6)$$

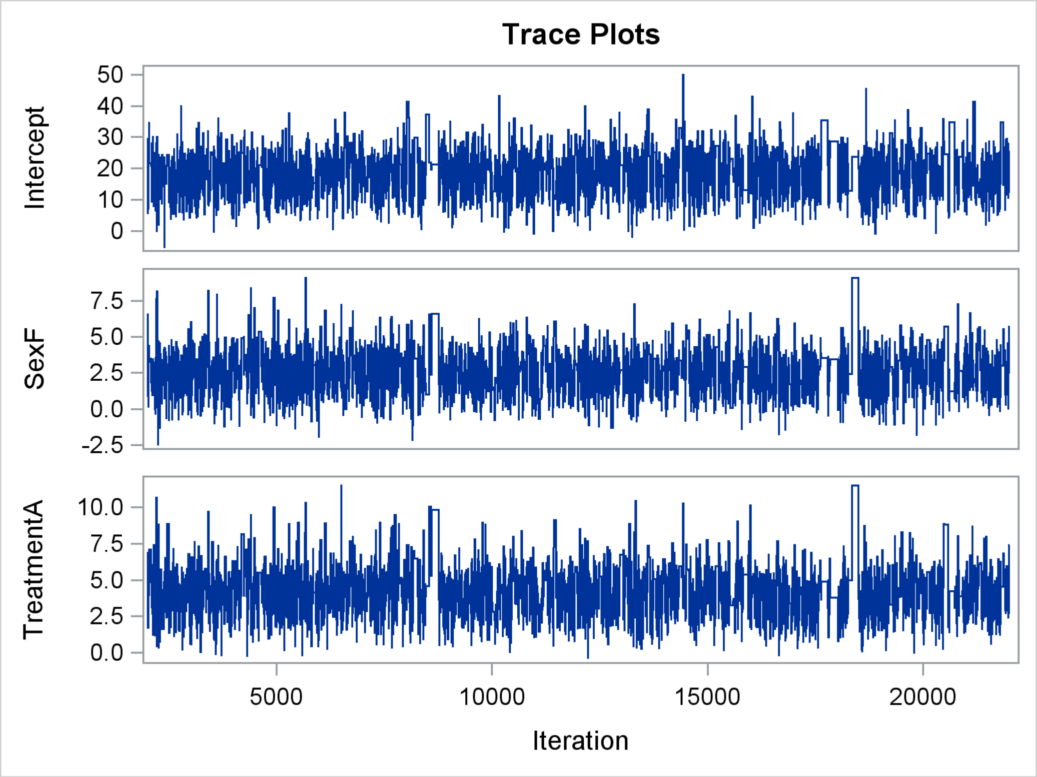
You can also fit this model with the GENMOD procedure. The following statements specify the analysis:

```
proc genmod data=neuralgia;
   class Treatment(ref="P") Sex(ref="M");
   model Pain= sex|treatment Age Duration / dist=bin link=logit;
   bayes seed=1 cprior=normal(var=1e6) outpost=neuout plots=trace nmc=20000;
run;
```

The CPRIOR=NORMAL(VAR=1E6) option specifies the normal prior for the coefficients; the specified large variance is requested with VAR=1E6. The PLOTS=TRACE option requests only the trace plots. Logistic regression is requested with the DIST=BIN and the LINK=LOGIT options (either one will do). The default sampling algorithm for generalized linear models is the Gamerman algorithm, which uses an iterative weighted least squares algorithm to sample the coefficients from their conditional distributions. It usually performs very well. The NMC option requests 20,000 samples and is specified because the default number of simulations results in low ESS.

Figure 13 displays the trace plots for some of the parameters. They show good mixing.

Figure 13 Trace Plots



The effective sample sizes are adequate and do not indicate any issues in the convergence of the Markov chain, as seen in Figure 14. With 20,000 samples, the chain has stabilized appropriately.

Figure 14 Effective Sample Sizes

Effective Sample Sizes

Parameter	ESS	Autocorrelation Time	Efficiency
Intercept	685.2	29.1890	0.0343
SexF	569.3	35.1325	0.0285
TreatmentA	401.8	49.7762	0.0201
TreatmentB	491.7	40.6781	0.0246
TreatmentASexF	596.4	33.5373	0.0298
TreatmentBSexF	585.2	34.1753	0.0293
Age	604.3	33.0943	0.0302
Duration	1054.7	18.9619	0.0527

Thus, the posterior summaries are of interest. Figure 15 displays the posterior summaries and Figure 16 displays the posterior intervals.

Figure 15 Posterior Summaries

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	 50%	 75%
Intercept	20000	19.6387	7.5273	14.2585	19.6414	24.4465
SexF	20000	2.7964	1.6907	1.6403	2.6644	3.7348
TreatmentA	20000	4.5857	1.9187	3.3402	4.4201	5.5466
TreatmentB	20000	5.0661	1.9526	3.7114	4.8921	6.2388
TreatmentASexF	20000	-1.0052	2.3403	-2.3789	-0.9513	0.5297
TreatmentBSexF	20000	-0.2559	2.2854	-1.7394	-0.2499	1.2874
Age	20000	-0.3365	0.1134	-0.4192	-0.3316	-0.2522
Duration	20000	0.00790	0.0375	-0.0164	0.00668	0.0312

Figure 16 Posterior Intervals

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	5.7037	35.4140	4.0078	32.9882
SexF	0.050	-0.0605	6.5737	0.0594	6.6094
TreatmentA	0.050	1.3776	9.4931	1.2502	8.9476
TreatmentB	0.050	1.6392	9.2298	1.7984	9.3168
TreatmentASexF	0.050	-5.8694	3.4826	-5.9435	3.2389
TreatmentBSexF	0.050	-4.7975	4.3188	-4.1485	4.4298
Age	0.050	-0.5721	-0.1338	-0.5422	-0.1159
Duration	0.050	-0.0637	0.0869	-0.0669	0.0821

The intervals suggest that treatment is highly influential, and although age appears to be important, other covariates and the sex × treatment interactions do not appear to be important. However, all terms are kept in the model.

Odds ratios are an important measure of association that can be estimated by forming functions of the parameter estimates for a logistic regression. See Stokes, Davis, and Koch (2012) for a full discussion. For example, if you want to form an odds ratio comparing the odds of no pain for the female patients to the odds of no pain for the male patients, you can compute it by exponentiating the corresponding linear combination of the relevant model parameters.

Because a function of random variables is also a random variable, you can compute odds ratios from the results of a Bayesian analysis by manipulating the posterior samples. You save the results of your analysis to a special data set known as a SAS item store using the STORE statement, and then you compute the odds ratio by using the ESTIMATE statement of the PLM procedure. PROC PLM performs postfitting tasks by operating on the posterior samples from PROC GENMOD. (For an MLE analysis, it operates on the saved parameter estimates and covariances.) Recall that the ESTIMATE statement provides custom hypotheses, which means that it can also be used to form functions from the linear combinations of the regression coefficients.

The following statements fit the same model with PROC GENMOD and saves the posterior samples to the LOGIT_BAYES item store:

```
proc genmod data=neuralgia;
   class Treatment(ref="P") Sex(ref="M");
   model Pain= sex|treatment Age Duration / dist=bin link=logit;
   bayes seed=2 cprior=normal(var=1e6) outpost=neuout
```

```
      plots=trace algorithm=im;
      store logit_bayes;
   run;
```

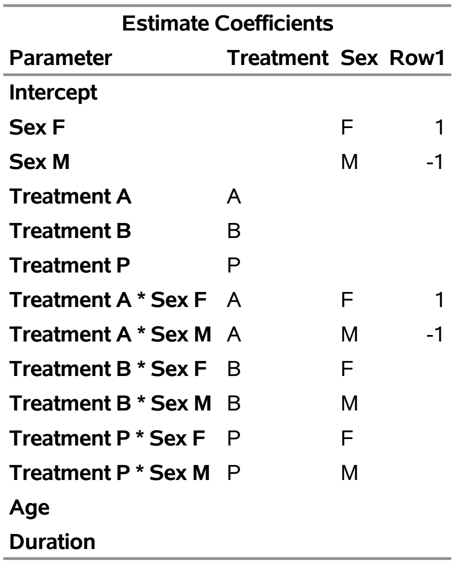
Note that a different sampler is requested with the SAMPLING=IM option, which requests the Independence Metropolis sampler, which is a variation of the Metropolis sampling algorithm. This sampler finds an envelope distribution to the posterior and uses it as an efficient proposal distribution to generate the posterior samples. It is generally faster than the Gamerman algorithm, but the latter is more computationally stable so it remains the default sampler in PROC GENMOD. In this case, the IM sampler produces better ESS values.

The next set of statements inputs the LOGIT_BAYES item store into the PLM procedure with the RESTORE= option. The desired odds ratio estimate is specified with the ESTIMATE statement. Because this model includes the SEX*TREATMENT interaction, the odds ratio requested is the one that compares female and male odds at the A treatment level. The 1 and -1 coefficients compare females and males for the overall gender effect and also at the A level for treatment. You are effectively creating a distribution of these functions from each of the posterior sample estimates. Plots are provided by the PLM procedure when it is executed in this sample-based mode, and the EXP option requests the exponentiation that produces the odds ratio estimate from the parameters. The CL option requests 95% credible intervals.

```
   proc plm restore=logit_bayes;
      estimate "F vs M, at Trt=A"
        sex 1 -1 treatment*sex 1 -1 0 0 0 0
      / exp cl;
   run;
```

Figure 17 displays the coefficients of the L matrix for confirmation that the appropriate contrast was specified.

Figure 17 L Matrix Coefficients



Estimate Coefficients

Parameter	Treatment	Sex	Row1
Intercept			
Sex F		F	1
Sex M		M	-1
Treatment A	A		
Treatment B	B		
Treatment P	P		
Treatment A * Sex F	A	F	1
Treatment A * Sex M	A	M	-1
Treatment B * Sex F	B	F	
Treatment B * Sex M	B	M	
Treatment P * Sex F	P	F	
Treatment P * Sex M	P	M	
Age			
Duration			

Figure 18 displays the odd ratio estimate in the “Exponentiated” column, which has the value 28.39. This says that the odds of no pain are 28 times higher for females than for males at the A treatment level.

Figure 18 Posterior Odds Ratio Estimate

					Sample Estimate						
					Percentiles						
Label	N	Estimate	Standard Deviation	25th	50th	75th	Alpha	Lower HPD	Upper HPD	Exponentiated	Standard Deviation of Exponentiated
F vs M, at Trt=A	10000	1.8781	1.5260	0.7768	1.7862	2.9174	0.05	-0.7442	4.9791	28.3873	188.824003

		Sample Estimate			
		Percentiles for Exponentiated			
Label	25th	50th	75th	Lower HPD of Exponentiated	Upper HPD of Exponentiated
F vs M, at Trt=A	2.1744	5.9664	18.4925	0.1876	93.1034

Proportional Hazards Model

Consider the data from the Veteran Adminstration's lung cancer trial that are presented in Appendix 1 of Kalbfleisch and Prentice (1980). Death in days is the response, and type of therapy, type of tumor cell, presence of prior therapy, age, and time from diagnosis to study inclusion are included in the data set. There is also an indicator variable for whether the response was censored.

The first 20 observations of SAS data set VALUNG are shown in Figure 19.

below:

Figure 19 Subset of VALUNG Data

Obs	Therapy	Cell	Time	Kps	Duration	Age	Ptherapy	Status
1	standard	squamous	72	60	7	69	no	1
2	standard	squamous	411	70	5	64	yes	1
3	standard	squamous	228	60	3	38	no	1
4	standard	squamous	126	60	9	63	yes	1
5	standard	squamous	118	70	11	65	yes	1
6	standard	squamous	10	20	5	49	no	1
7	standard	squamous	82	40	10	69	yes	1
8	standard	squamous	110	80	29	68	no	1
9	standard	squamous	314	50	18	43	no	1
10	standard	squamous	100	70	6	70	no	0
11	standard	squamous	42	60	4	81	no	1
12	standard	squamous	8	40	58	63	yes	1
13	standard	squamous	144	30	4	63	no	1
14	standard	squamous	25	80	9	52	yes	0
15	standard	squamous	11	70	11	48	yes	1
16	standard	small	30	60	3	61	no	1
17	standard	small	384	60	9	42	no	1
18	standard	small	4	40	2	35	no	1
19	standard	small	54	80	4	63	yes	1
20	standard	small	13	60	4	56	no	1

Interest lies in how therapy and the other covariates impact time until death, and the analysis is further complicated because the response is censored. The proportional hazards model, or Cox regression model, is widely used in the analysis of time-to-event data to explain the effect of explanatory variables on hazard rates. The survival time of each member of a population is assumed to follow its own hazard function $\lambda_i(t)$,

$$\lambda_l(t) = \lambda(t; Z_l) = \lambda_0(t)\exp(Z_l'\beta)$$

where $\lambda_0(t)$ is an arbitrary and unspecified baseline hazard function, Z_l is the vector of explanatory variables for the lth individual, and β is the vector of unknown regression parameters (which is assumed to be the same for all individuals). The PHREG procedure fits the Cox model by maximizing the partial likelihood function; this eliminates the unknown baseline hazard $\lambda_0(t)$ and accounts for censored survival times. In the Bayesian approach, the partial likelihood function is used as the likelihood function in the posterior distribution (Sinha, Ibrahim, and Chen 2003).

The PHREG procedure supports the following priors for proportional hazards regression, displayed in Table 3.

Table 3 Prior Distributions for the Cox Model

Parameter	Prior
Regression coefficients	Normal, uniform, Zellner's g-prior
Zellner's g	Constant, gamma

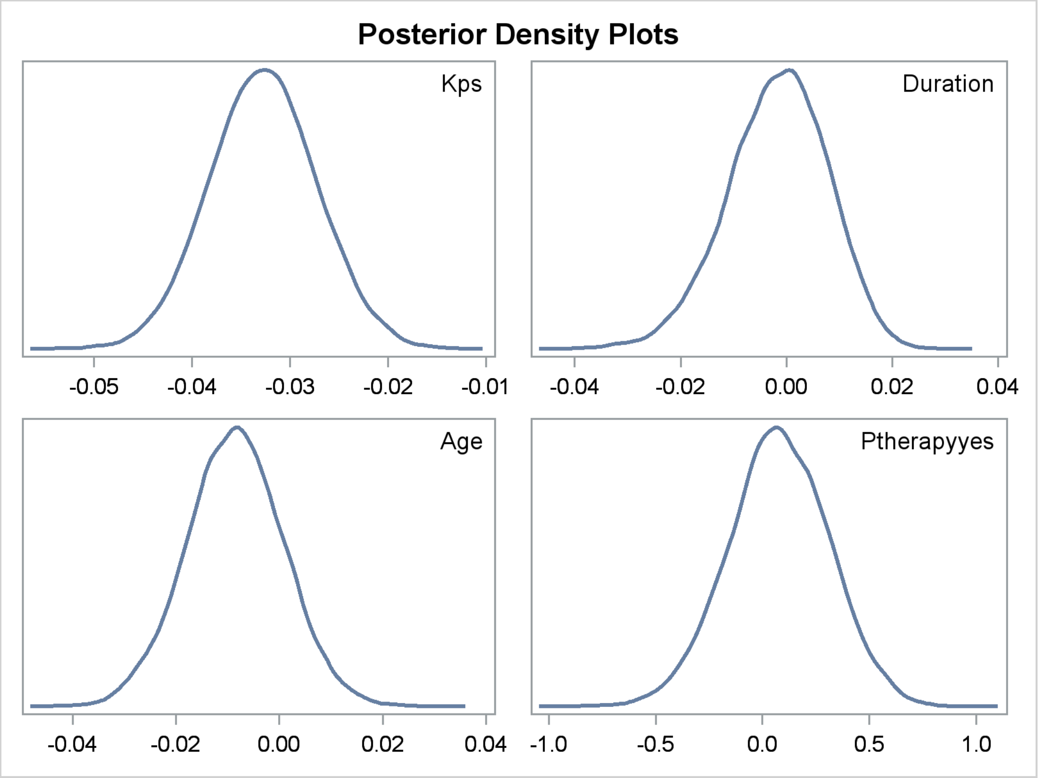
The following PROC PHREG statements request Bayesian analysis for the Cox proportional hazard model with uniform priors on the regression coefficients. The categorical variables PTHERAPY, CELL, and THERAPY are declared in the CLASS statement. The value “large” is specified as the reference category for type of tumor cell, the value “standard” is specified as the reference category for therapy, and the value “no” is the reference category for prior therapy.

```
   proc phreg data=VALung;
    class Ptherapy(ref='no') Cell(ref='large') Therapy(ref='standard');
    model Time*Status(0) = Kps Duration Age Ptherapy Cell Therapy;
    hazardratio 'HR' Therapy;
    bayes seed=1 outpost=out cprior=uniform plots=density;
 run;
```

The BAYES statement requests Bayesian analysis. The CPRIOR= option specifies the uniform prior for the regression coefficients, and the PLOTS= option requests a panel of posterior density plots for the model parameters. The HAZARDRATIO statement requests the hazard ratio comparing the odds for therapies.

Figure 20 shows kernel posterior density plots that estimate the posterior marginal distributions for the first four regression coefficients. Each plot shows a smooth, unimodal shape for the posterior marginal distribution.

Figure 20 Density Plots



Posterior summary statistics and intervals are shown in Figure 21. Because the prior distribution is noninformative, the results are similar to those obtained with the standard PROC PHREG analysis (not shown here).

Figure 21 Posterior Summary Statistics

Bayesian Analysis

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Kps	10000	-0.0327	0.00545	-0.0434	-0.0221
Duration	10000	-0.00170	0.00945	-0.0202	0.0164
Age	10000	-0.00852	0.00935	-0.0270	0.00983
Ptherapyyes	10000	0.0754	0.2345	-0.3715	0.5488
Celladeno	10000	0.7867	0.3080	0.1579	1.3587
Cellsmall	10000	0.4632	0.2731	-0.0530	1.0118
Cellsquamous	10000	-0.4022	0.2843	-0.9550	0.1582
Therapytest	10000	0.2897	0.2091	-0.1144	0.6987

The posterior mean of the hazard ratio in Figure 22 indicates that the standard treatment is more effective than the test treatment for individuals who have the given covariates. However, note that both the 95% equal-tailed interval and the 95% HPD interval contain the hazard ratio value of 1, which suggests that the test therapy is not really different from the standard therapy.

Figure 22 Hazard Ratio and Confidence Limits

HR: Hazard Ratios for Therapy

				Quantiles						
			Standard				95% Equal-Tail		95%	
Description	N	Mean	Deviation	25%	50%	75%	Interval		HPD Interval	
Therapy standard vs test	10000	0.7651	0.1617	0.6509	0.7483	0.8607	0.4988	1.1265	0.4692	1.0859

Suppose you are interested in estimating the survival curves for two individuals who have similar characteristics, one of whom receives the test treatment while the other receives the standard treatment. The following SAS DATA step creates the PRED data set which includes the covariate values for these individuals:

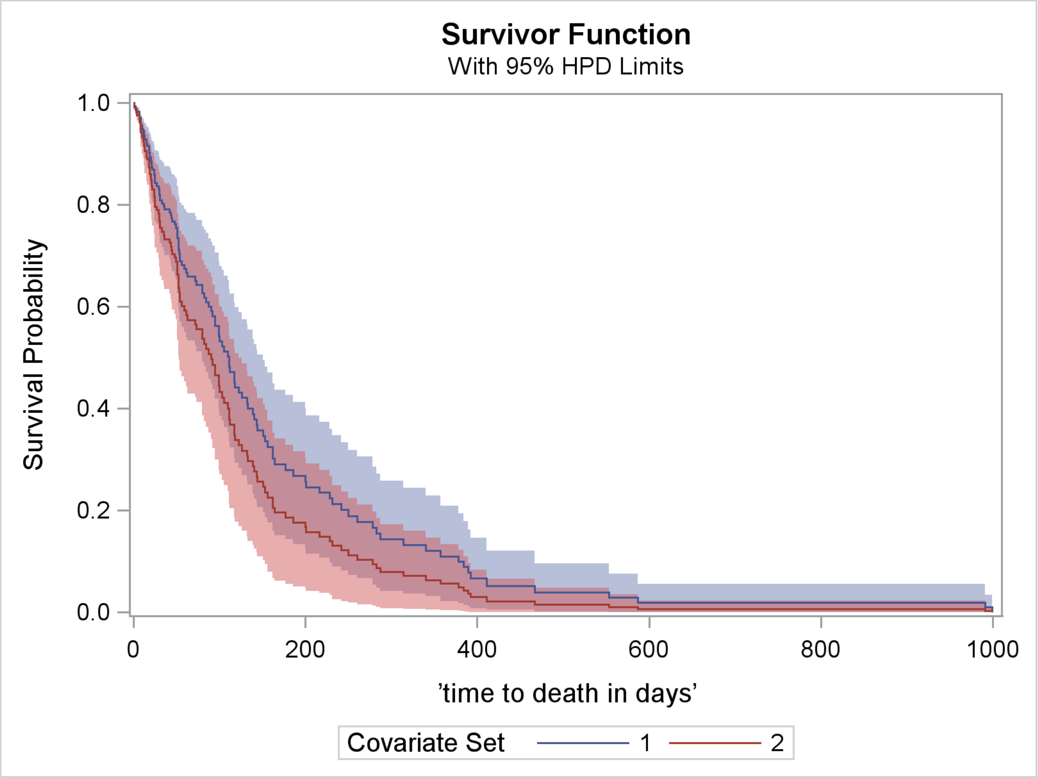
```
data Pred;
   input Ptherapy  Kps Duration Age Cell $ Therapy $ @@;
   format Ptherapy yesno.;
   datalines;
   0 58 8.7 60 large standard
   0 58 8.7 60 large test
   ;
```

The following statements request the estimation of separate survival curves for these two sets of covariates. The PLOTS= option in the PROC PHREG statement requests survival curves, the OVERLAY suboption overlays the curves in the same plot, and the CL= suboption requests the highest posterior density (HPD) confidence limits for the survivor functions. The COVARIATES= option in the BASELINE statement specifies the covariates data set PRED.

```
proc phreg data=VALung plots(overlay cl=hpd)=survival;
   baseline covariates=Pred;
   class Therapy(ref='standard') Cell(ref='large') Ptherapy(ref='no');
   model Time*Status(0) = Kps Duration Age Ptherapy Cell Therapy;
   bayes seed=1 outpost=out cprior=uniform plots=density;
run;
```

Figure 23 displays the survival curves for these sets of covariate values along with their HPD confidence limits.

Figure 23 Posterior Survival Curves



Capabilities of SAS Built-In Bayesian Procedures

The examples in this paper provide an introduction to the capabilities of the built-in Bayesian procedures in SAS/STAT. But not only can the GENMOD procedure produce Bayesian analyses for linear regression and logistic regression, it also provides Bayesian analysis for Poisson regression, negative binomial regression, and models based on the Tweedie distribution. While PROC GENMOD doesn't offer Bayesian analysis for zero-inflated models, such as the zero-inflated Poisson model, you can still use the built-in Bayesian procedures by switching to the FMM procedure, which fits finite mixture models, which includes zero-inflated models.

The FMM procedure provides Bayesian analysis for finite mixtures of normal, T, binomial, binary, Poisson, and exponential distributions. This covers ground from zero-inflated models to clustering for univariate data. PROC FMM applies specialized sampling algorithms, depending on the distribution family and the nature of the model effects, including conjugate samplers and the Metropolis-Hastings algorithm. PROC FMM provides a single parametric form for the prior but you can adjust it to reflect specific prior information. It provides posterior summaries and convergence diagnostics similar to those produced by the other built-in Bayesian procedures. See Kessler and McDowell (2012) for an example of Bayesian analysis using the FMM procedure.

This paper illustrates the use of the PHREG procedure to perform Bayesian analysis for Cox regression; it can handle time-dependent covariates and all TIES= methods. (The Bayesian functionality does not currently apply to models that have certain data constraints such as recurrent events.) Bayesian analysis is also available for the frailty models fit with the PHREG procedure, which enable you to analyze survival times that are clustered. In addition, PROC PHREG also provides Bayesian analysis for the piecewise exponential model.

The LIFEREG procedure provides Bayesian analysis for parametric lifetime models. This capability is now available for the exponential, 3-parameter gamma, log-logistic, log-normal, logistic, normal, and Weibull distributions. Model parameters are the regression coefficients and dispersion parameters (or precision or scale). Normal and uniform priors are provided for the regression coefficients, and the gamma and improper priors are provided for the scale and shape parameters.

General Bayesian Modeling with the MCMC Procedure

Although the built-in Bayesian procedures provide Bayesian analyses for many standard techniques, they only go so far. You might want to include priors that are not offered, or you might want to perform a Bayesian analysis for a model that isn't covered. For example, the GENMOD procedure doesn't presently offer Bayesian analysis for the proportional odds model. However, you can fit nearly any model that you want, for any prior and likelihood you can program, with the MCMC procedure.

The MCMC procedure is a general-purpose Bayesian modeling tool. It was built on the familiar syntax of programming statements used in the NLMIXED procedure. PROC MCMC performs posterior sampling and statistical inference for Bayesian parametric models. The procedure fits single-level or multilevel models. These models can take various forms, from linear to nonlinear models, by using standard or nonstandard distributions. Using PROC MCMC requires using programming statement to declare the parameters in the model, to specify prior distributions for them, and to describe the conditional distribution for the response variable given the parameters. You can specify a model by using keywords for a standard form (normal, binomial, gamma) or use programming statements to define a general distribution.

The MCMC procedure uses various algorithms to simulate samples from the model that you specify. You can also choose an optimization technique (such as the quasi-Newton algorithm) to estimate the posterior mode and approximate the covariance matrix around the mode. PROC MCMC computes a number of posterior estimates, and it also outputs the posterior samples to a data set for further analysis. The MCMC procedure provides the same convergence diagnostics that are produced by using a BAYES statement in the built-in Bayesian procedures, as discussed previously.

The RANDOM statement in the MCMC procedure facilitates the specification of random effects in hierarchical linear or nonlinear models. You can build nested or nonnested hierarchical models to arbitrary depth. Using the RANDOM statement can result in reduced simulation time and improved convergence for models that have a large number of subjects. The MCMC procedure also handles missing data for both responses and covariates. Chen (2009, 2011, 2013) provides good overviews of the capabilities of the MCMC procedure, which continues to be enhanced. The most recent release of PROC MCMC in SAS/STAT 13.1 is multithreaded, providing faster performance for many models.

The BCHOICE Procedure

The first SAS/STAT procedure designed to focus on Bayesian analysis for a specific application is the BCHOICE procedure. PROC BCHOICE performs Bayesian analysis for discrete choice models. Discrete choice models are used in marketing research to model decision makers' choices among alternative products and services. The decision makers might be people, households, companies, and so on, and the alternatives might be products, services, actions, or any other options or items about which choices must be made. The collection of alternatives is known as a choice set, and when individuals are asked to choose, they usually assign a utility to each alternative. The BCHOICE procedure provides Bayesian discrete choice models such as the multinomial logit, multinomial logit with random effects, and the nested logit. The probit response function is also available. You can supply a prior distribution for the parameters if you want something other than the default noninformative prior. PROC BCHOICE obtains samples from the corresponding posterior distributions, produces summary and diagnostic statistics, and saves the posterior samples to an output data set that can be used for further analysis. PROC BCHOICE is also multithreaded.

Summary

The SAS built-in Bayesian procedures provide a great deal of coverage for standard statistical analyses. With a ready set of priors and carefully-chosen default samplers, they make Bayesian computing very convenient for the SAS/STAT user. They provide a great starting place for the statistician who needs some essential Bayesian analysis now and might want to add more sophisticated Bayesian computing skills later. Bayesian analysis is an active area in SAS statistical software development, and each new release brings additional features.

For more information, the “Introduction to Bayesian Analysis” chapter in the *SAS/STAT User's Guide* contains a reading list with comprehensive references, including many references at the introductory level. In addition, the Statistics and Operations Focus Area includes substantial information about the statistical products, and you can find it at support.sas.com/statistics/. The quarterly e-newsletter for that site is available on its home page. And of course, complete information is available in the online documentation located at support.sas.com/documentation/onlinedoc/stat/. The features of each new release are described in a “What’s New” chapter that is must reading for SAS/STAT users.

References

Chen, F. (2009), “Bayesian Modeling Using the MCMC Procedure,” in *Proceedings of the SAS Global Forum 2008 Conference,* Cary NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings09/257-2009.pdf.

Chen, F. (2011), “The RANDOM Statement and More: Moving on with PROC MCMC,” in *Proceedings of the SAS Global Forum 2011 Conference,* Cary NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/334-2011.pdf.

Chen, F. (2013), “Missing No More: Using the MCMC Procedure to Model Missing Data”, in *Proceedings of the SAS Global Forum 2013 Conference,* Cary NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings11/436-2013.pdf.

Gamerman, D. (1997), “Sampling from the Posterior Distribution in Generalized Linear Mixed Models,”*Statistics and Computing*, 7, 57–68.

Gilks, W. R. and Wild, P. (1992), “Adaptive Rejection Sampling for Gibbs Sampling,” *Applied Statistics*, 41, 337–348.

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995), “Adaptive Rejection Metropolis Sampling,” *Applied Statistics*, 44, 455–472.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons.

Kessler, D. and McDowell, A. (2012), “Introducing the FMM Procedure for Finite Mixture Models,” in *Proceedings of the SAS Global Forum 2012 Conference,* Cary NC: SAS Institute Inc. Available at http://support.sas.com/resources/papers/proceedings12/328-2012.pdf.

Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. (1996), *Applied Linear Statistical Models,* Fourth Edition, Chicago: Irwin.

Sinha, D., Ibrahim, J. G., and Chen, M. (2003), “Bayesian Justification of Cox’s Partial Likelihood”, *Biometrics*, 90, 629–641.

Stokes, M. E., Davis, C. E., and Koch G G. (2012). *Categorical Data Analysis Using SAS, Third Edition*, SAS Press: Cary NC.

Acknowledgments

The authors are grateful to Ed Huddleston for editing this manuscript. They would also like to thank David Kessler and Warren Kuhfeld for their helpful suggestions.

Contact Information

Your comments and questions are valued and encouraged. Contact the author:

Maura Stokes
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513

Version

1.0

Paper SAS3042-2019

Introducing the BGLIMM Procedure for Bayesian Generalized Linear Mixed Models

Amy Shi and Fang Chen, SAS Institute Inc., Cary, NC

ABSTRACT

SAS/STAT® 15.1 includes PROC BGLIMM, a new, high-performance, sampling-based procedure that provides full Bayesian inference for generalized linear mixed models. PROC BGLIMM models data from exponential family distributions that have correlations or nonconstant variability; uses syntax similar to that of the MIXED and GLIMMIX procedures (the CLASS, MODEL, RANDOM, REPEATED, and ESTIMATE statements); deploys optimal sampling algorithms that are parallelized for performance; handles multilevel nested and non-nested random-effects models; and fits models to multivariate or longitudinal data that contain repeated measurements. PROC BGLIMM provides convenient access, with improved performance, to Bayesian analysis of complex mixed models that you could previously perform with the MCMC procedure. This paper describes how to use the BGLIMM procedure for estimation, inference, and prediction.

INTRODUCTION

A generalized linear mixed model (GLMM) is an extension of the generalized linear model (GLM) in which the linear predictor contains random effects in addition to the usual fixed effects. GLMMs also inherit from GLMs the idea of extending linear mixed models to nonnormal data. Conditional on the random effects, data have distributions in the exponential family (binary, binomial, Poisson, normal, gamma, and so on). GLMMs are widely used in practice and are especially useful in applications where the data consist of collections of units and are hierarchically structured.

The popular MIXED and GLIMMIX procedures fit GLMM models by the classical approach of maximizing a marginal likelihood function (integrated over the random effects) to estimate model parameters. PROC BGLIMM instead takes a Bayesian approach, using simulation techniques to draw samples from the joint posterior distribution of all model parameters and then using these samples to estimate and infer on quantities of interest. The direct estimation of the parameters' posterior distribution, although computationally expensive, is an essential feature of Bayesian inference, and it bypasses the dependency on asymptotic sampling distributions that is required by likelihood-based inference.

PROC BGLIMM uses a variety of sampling algorithms to draw samples from the posterior distribution of parameters. These algorithms include the conjugate sampler, direct sampler, Gamerman algorithm (a variation of the Metropolis-Hastings algorithm that is tailored to generalized linear models; see Gamerman 1997), and No-U-Turn Sampler (NUTS, a self-tuning variation of the Hamiltonian Monte Carlo (HMC) method; see Neal 2011 and Hoffman and Gelman 2014). The algorithms are parallelized to reduce run time.

Successful convergence of the Markov chain results in precise estimation of the posterior distribution (which can be summarized using point and interval estimates) that you can use to quantify uncertainties about the model parameters. PROC BGLIMM estimates linear functions of model parameters directly (via the ESTIMATE statement), and you can use the posterior samples to carry out additional posterior inferences or further analysis.

In terms of syntax, PROC BGLIMM adheres to the tradition that PROC MIXED and PROC GLIMMIX established, with similar CLASS, MODEL, RANDOM, REPEATED, and ESTIMATE statements. This provides an easy transition for SAS users who are familiar with the established conventions.

The paper is organized as follows. "Notation" provides a brief overview of GLMMs. "The BGLIMM Procedure" introduces important features, statements, and options in PROC BGLIMM. "BGLIMM Procedure Details" covers high-level simulation and algorithm details of the procedure. "Prior Distributions" discusses prior specification. "Examples" presents three examples, from simple to complex, to demonstrate how to use the procedure.

NOTATION

First consider the normal linear mixed model. The quantity of primary interest, y_i, is called the response or outcome variable for the ith individual. The distribution of y_i is normal,

$$\begin{aligned} y_i &= \mathbf{x}_i \boldsymbol{\beta} + \mathbf{z}_i \boldsymbol{\gamma}_i + \epsilon_i, \quad i = 1, \ldots, I \\ \boldsymbol{\gamma}_i &\sim N(\mathbf{0}, \mathbf{G}_i) \\ \epsilon_i &\sim N(0, \mathbf{R}_i) \end{aligned}$$

where $\boldsymbol{\beta}$ is a $p \times 1$ vector of fixed effects, $\boldsymbol{\gamma}_i$ is a $q \times 1$ vector of random effects, ϵ_i is the normal noise with a variance $\mathbf{R}_i = \sigma^2$, and $\mathbf{G}_i$ is the covariance matrix of the random effects $\boldsymbol{\gamma}_i$ ($\mathbf{G}$ is a block diagonal matrix where each block is $\mathbf{G}_i$).

When an individual i has n_i repeated measurements, the random-effects model for outcome vector $\mathbf{y}_i$ is given by

$$\mathbf{y}_i = \mathbf{X}_i \boldsymbol{\beta} + \mathbf{Z}_i \boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i$$

where $\mathbf{y}_i$ is $n_i \times 1$, $\mathbf{X}_i$ is an $n_i \times p$ design matrix of fixed covariates, $\boldsymbol{\beta}$ is a $p \times 1$ vector of fixed effects, $\boldsymbol{\gamma}_i$ is a $q \times 1$ vector of random effects, $\mathbf{Z}_i$ is an $n_i \times q$ design matrix of covariates for the $\boldsymbol{\gamma}_i$, and $\boldsymbol{\epsilon}_i$ is an $n_i \times 1$ vector of random errors. $\mathbf{R}_i$ is the covariance matrix of the residual errors for the ith subject ($\mathbf{R}$ is a block diagonal matrix where each block is $\mathbf{R}_i$).

There are cases where the relationship between the design matrices and the expectation of the response is not linear, or where the distribution for the response is far from normal, even after the data are transformed. The class of GLMMs unifies the approaches that you need in order to analyze data in those cases. Let $\mathbf{Y}$ be the collection of all $\mathbf{y}_i$, and let $\mathbf{X}$ and $\mathbf{Z}$ be the collection of all $\mathbf{X}_i$ and $\mathbf{Z}_i$, respectively. A GLMM model consists of the following:

- the linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$
- the link function $g(\cdot)$ that relates the linear predictor to the mean of the outcome via a monotone link function,

  $$\mathrm{E}[Y|\boldsymbol{\beta}, \boldsymbol{\gamma}] = g^{-1}(\boldsymbol{\eta}) = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma})$$

  where $g(\cdot)$ is a differentiable monotone link function and $g^{-1}(\cdot)$ is its inverse
- a response distribution in the exponential family of distributions. The distribution can also depend on a scale parameter, ϕ.

The conditional distribution of the response variable, given $\boldsymbol{\gamma}$, is a member of the exponential family of distributions (binary, binomial, Poisson, normal, gamma, and so on).

There are two types of covariance structures: the "G-side" and the "R-side." The G-side matrix, $\mathbf{G}$, is the covariance matrix of the random effects; the R-side matrix, $\mathbf{R}$, is the covariance matrix of the residuals. By default, the $\mathbf{R}$ matrix is the scaled identity matrix, $\mathbf{R} = \phi\mathbf{I}$, where the scale parameter ϕ is set to 1 if the distribution does not have a scale parameter, such as in the case of the binary, binomial, Poisson, and exponential distributions. Models without G-side effects are also known as marginal (or population-averaged) models.

THE BGLIMM PROCEDURE

PROC BGLIMM provides the following features:

- nested or non-nested hierarchical models
- repeated-measures models (balanced or unbalanced data) with normal data
- suite of covariance structures for random effects and residuals, including variance components, compound symmetry, unstructured, AR(1), Toeplitz, autoregressive, and many more
- built-in prior distributions for regression coefficients and covariance parameters
- the ability to model heterogeneity in covariance structures

- the ability to produce estimates and credible intervals for estimable linear combination of effects
- support for missing completely at random (MCAR) and missing at random (MAR) approaches in modeling missing data
- multithreading of optimal sampling algorithms for fast performance
- the ability to save posterior samples to an output data set for use in further inferences

PROC BGLIMM uses syntax similar to that of PROC MIXED and PROC GLIMMIX in specifying a GLMM. The following three statements are the most essential:

- MODEL statement: specifies the response variable, fixed effects, likelihood function (DIST= option), and link function (LINK= option)
- RANDOM statement: specifies the random effects and the G-side variance or covariance structure (TYPE= option)
- REPEATED statement: specifies the R-side residual variance or covariance structure (TYPE= option)

More detailed descriptions of the three statements follow.

MODEL *response = fixed-effects* < / model-options>;

The MODEL statement specifies the dependent variable and fixed-effects parameters. You can also use this statement to specify the response distribution via the DIST= option and to specify the link function $g(\cdot)$ via the LINK= option. Some other useful options follow:

- NOINT excludes the fixed-effects intercept from the model.
- OFFSET= specifies the offset variable.
- COEFFPRIOR= specifies the prior of the fixed-effects coefficients.
- SCALEPRIOR= specifies the prior of the scale parameter.

You can use PROC BGLIMM to fit the likelihood functions that are listed in Table 1.

Table 1 Built-In Distribution Functions

DIST= Option Value	**Distribution Function**
BINARY	Binary
BINOMIAL	Binary or binomial
EXPONENTIAL | **EXPO**	Exponential
GAMMA | **GAM**	Gamma
GEOMETRIC | **GEOM**	Geometric
INVGAUSS | **IG**	Inverse Gaussian
NEGBINOMIAL | **NEGBIN** | **NB**	Negative binomial
NORMAL | **GAUSSIAN** | **GAUSS**	Normal
POISSON | **POI**	Poisson

The default distribution is normal for continuous variable and binomial for categorical variables. Supported link functions are shown in Table 2, and the default and other commonly used link functions for the available distributions are listed in Table 3.

Table 2 Built-In Link Functions

LINK=	Link Function	$g(\mu) = \eta =$
CLOGLOG | CLL	Complementary log-log	$\log(-\log(1-\mu))$
IDENTITY | ID	Identity	μ
INVERSE | RECIPROCAL	Reciprocal	$1/\mu$
LOG	Logarithm	$\log(\mu)$
LOGIT	Logit	$\log(\mu/(1-\mu))$
LOGLOG	Log-log	$-\log(-\log(\mu))$
POWERMINUS2	Power with exponent –2	$1/\mu^2$
PROBIT	Probit	$\Phi^{-1}(\mu)$

Table 3 Default and Commonly Used Link Functions

DIST= Option Value	Default Link Function	Other Commonly Used Link Functions
BINARY	Logit	Probit, complementary log-log, log-log
BINOMIAL	Logit	Probit, complementary log-log, log-log
EXPONENTIAL | EXPO	Log	Reciprocal
GAMMA | GAM	Log	Reciprocal
GEOMETRIC | GEOM	Log	
INVGAUSS | IG	Reciprocal square	
NEGBINOMIAL | NEGBIN | NB	Log	
NORMAL | GAUSSIAN | GAUSS	Identity	Log
POISSON | POI	Log	

RANDOM *random-effects* < / options>;

The RANDOM statement defines the **Z** matrix of the mixed model, the random effects in the $\boldsymbol{\gamma}$ vector, and the covariance structure of the **G** matrix. You specify the SUBJECT= option to identify the subjects for the random effects and thus to set up the blocks of **G**. A set of random effects is estimated for each subject level. You define the covariance structure of **G** by using the TYPE= option. The random effects can be classification or continuous effects, and you can specify multiple RANDOM statements. You can also specify the GROUP= option to identify groups by which to vary the covariance parameters; each new level of the grouping effect produces a new set of covariance parameters.

You can specify INTERCEPT (or INT) as a random effect to indicate the intercept. PROC BGLIMM does not include the intercept in the RANDOM statement by default as it does in the MODEL statement.

Table 4 lists the supported **G**-matrix covariance types. The default is TYPE=VC.

Table 4 Covariance Structures

Structure	Description	Parms	(i,j) Element
ANTE(1)	Antedependence	$2t-1$	$\sigma_i \sigma_j \prod_{k=i}^{j-1} \rho_k$
AR(1)	Autoregressive(1)	2	$\sigma^2 \rho^{\lvert i-j \rvert}$
ARH(1)	Heterogeneous AR(1)	$t+1$	$\sigma_i \sigma_j \rho^{\lvert i-j \rvert}$
ARMA(1,1)	ARMA(1,1)	3	$\sigma^2[\gamma\rho^{\lvert i-j \rvert-1} 1(i \neq j) + 1(i = j)]$
CS	Compound symmetry	2	$\sigma_1 + \sigma^2 1(i = j)$
CSH	Heterogeneous compound symmetry	$t+1$	$\sigma_i \sigma_j [\rho 1(i \neq j) + 1(i = j)]$
FA(1)	Factor analytic	$2t$	$\Sigma_{k=1}^{\min(i,j,1)} \lambda_{ik}\lambda_{jk} + d_i 1(i = j)$
HF	Huynh-Feldt	$t+1$	$(\sigma_i^2 + \sigma_j^2)/2 - \lambda 1(i \neq j)$

Table 4 *continued*

Structure	Description	Parms	(i, j) Element
TOEP	Toeplitz	t	$\sigma_{\|i-j\|+1}$
TOEP(q)	Banded Toeplitz	q	$\sigma_{\|i-j\|+1} 1(\|i-j\| < q)$
TOEPH	Heterogeneous Toeplitz	$2t-1$	$\sigma_i \sigma_j \rho_{\|i-j\|}$
TOEPH(q)	Banded heterogeneous Toeplitz	$t+q-1$	$\sigma_i \sigma_j \rho_{\|i-j\|} 1(\|i-j\| < q)$
UN	Unstructured	$t(t+1)/2$	σ_{ij}
UN(q)	Banded	$\frac{q}{2}(2t-q+1)$	$\sigma_{ij} 1(\|i-j\| < q)$
VC	Variance components	q	$\sigma_k^2 1(i = j)$ and i corresponds to the kth effect

In Table 4, Parms refers to the number of covariance parameters in the structure, t is the overall dimension of the covariance matrix, q is the order parameter, and $1(A)$ equals 1 when A is true and 0 otherwise.

REPEATED *repeated-effect* < / options>;

The REPEATED statement specifies the **R** matrix in the model. Its syntax is similar to that of the REPEATED statement in PROC MIXED. If you omit this statement, **R** is assumed to be equal to $\sigma^2\mathbf{I}$. The REPEATED statement is available only for the normal distribution with the identity link in this release.

Specifying a *repeated-effect* is required in order to inform PROC BGLIMM of the proper location of the observed repeated responses. The *repeated-effect* must contain only classification variables. You specify the SUBJECT= option to set up the blocks of **R**. You can use the TYPE= option to define the covariance structure. The levels of the *repeated-effect* must be different for each observation within a subject; otherwise, PROC BGLIMM produces an error message.

The same collection of covariance types (Table 4) is supported in the **R** matrix. Again, the default is TYPE=VC.

Descriptions of several more useful options and statements follow.

PROC BGLIMM options;

The PROC BGLIMM statement invokes the procedure. It includes these commonly used options:

- DATA= names the input data set.
- DIC computes the deviance information criterion.
- NBI= specifies the number of burn-in iterations.
- NMC= specifies the number of iterations, excluding the burn-in iterations.
- OUTPOST= names the output data set to contain posterior samples.
- SEED= specifies the random seed for simulation.
- STATS= controls posterior statistics.

BY variable(s);

You can specify a BY statement in PROC BGLIMM to obtain separate analysis of observations in groups that are defined by the BY variables.

CLASS variable(s);

The CLASS statement names the classification variables to be used in the model. You do not need to specify the response variable in the CLASS statement if it is categorical. The CLASS statement must precede the MODEL statement. You can specify the parameterization method for the classification variables—for example, the effect or reference coding scheme.

ESTIMATE 'label' *estimate-specification* < / options>;

The ESTIMATE statement enables you to compute a custom linear combination of the parameters. PROC BGLIMM produces for $\mathbf{L}'\boldsymbol{\phi}$, where $\boldsymbol{\phi}' = (\boldsymbol{\beta}'\,\boldsymbol{\gamma}')$, an estimate (by using the posterior mean), the standard deviation (by using the posterior standard deviation), and the highest posterior density (HPD) intervals.

BGLIMM PROCEDURE DETAILS

PROC BGLIMM updates parameters conditionally, through Gibbs sampling. The fixed-effects parameters $\boldsymbol{\beta}$ are drawn jointly at each iteration, the G-side and R-side covariance parameters are updated separately, and the random-effect parameters are updated by clusters. If you omit the SUBJECT= option from a RANDOM statement, all random-effects parameters from that statement are updated jointly (see the procedure documentation for more information about how the random-effects parameters can be parameterized with or without the presence of the SUBJECT= variable). Missing response values are treated as parameters and are thus sampled along with the other parameters mentioned earlier. Each missing response value is updated by using the likelihood function as the sampling distribution, conditional on the other parameters.

Conditional Distributions

Let $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \mathbf{G}, \mathbf{R}\}$, the collection of all fixed-effects parameters and the covariance matrices; let $\boldsymbol{\gamma}$ denote random-effects parameters, and let $\boldsymbol{\gamma}_j$ denote the random-effects parameters from cluster j. The treatment of random effects is identical for effects in multiple RANDOM statements.

The conditional distribution of $\boldsymbol{\beta}$ is

$$\log(p(\boldsymbol{\beta}|\boldsymbol{\gamma}, \mathrm{y}, \mathbf{R})) = \log(\pi(\boldsymbol{\beta})) + \sum_{i=1}^{n} \log(f(\mathrm{y}_i|\boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{R}))$$

where the log-likelihood function now includes the random effects $\boldsymbol{\gamma}$. This construction reflects two PROC BGLIMM modeling settings: one in which all random-effects parameters enter the likelihood function (linearly at the mean level), and one in which the fixed-effects parameters cannot be hyperparameters of $\boldsymbol{\gamma}$ (hence no $\log(\pi(\gamma_j|\boldsymbol{\beta}))$ terms).

The conditional distribution of $\mathbf{R}$ mirrors that of $\boldsymbol{\beta}$:

$$\log(p(\mathbf{R}|\boldsymbol{\gamma}, \mathrm{y}, \boldsymbol{\beta})) = \log(\pi(\mathbf{R})) + \sum_{i=1}^{n} \log(f(\mathrm{y}_i|\boldsymbol{\beta}, \boldsymbol{\gamma}, \mathbf{R}))$$

For $\boldsymbol{\gamma}_j$, the following conditional is used:

$$\log(p(\boldsymbol{\gamma}_j|\boldsymbol{\theta}, \mathrm{y})) = \log(\pi(\boldsymbol{\gamma}_j|\mathbf{G})) + \sum_{i \in \{j\text{th cluster}\}} \log(f(\mathrm{y}_i|\boldsymbol{\beta}, \boldsymbol{\gamma}_j, \mathbf{R}))$$

This computation uses only subjects from the jth cluster. This reflects the conditional independence assumption that the RANDOM statement makes. This simplification in the calculation makes updating the random-effects parameters computationally efficient and enables PROC BGLIMM to handle random effects that contain large numbers of clusters just as easily.

The G-side covariance matrix **G** depends only on the random effects γ and not on the data or other parameters, $\boldsymbol{\beta}$ or **R**,

$$\log(p(\mathbf{G}|\boldsymbol{\gamma})) = \log(\pi(\mathbf{G}))) + \sum_{j} \log(\pi(\boldsymbol{\gamma}_j|\mathbf{G}))$$

where $\pi(\mathbf{G})$ is the prior distribution of **G**.

Missing Values

PROC BGLIMM treats missing response values as parameters by default and samples them in the simulation. This mechanism of modeling missing values is referred to as the missing at random (MAR) approach. You can delete all observations that contain missing values by using the MISSING=CC option in the PROC BGLIMM statement.

Suppose that

$$\mathbf{y} = \{\mathbf{y_{obs}}, \mathbf{y_{mis}}\}$$

The response variable $\mathbf{y}$ consists of n_1 observed values, $\mathbf{y_{obs}}$, and n_2 missing values, $\mathbf{y_{mis}}$. At each iteration, PROC BGLIMM samples every missing response value (by using the likelihood function as the sampling distribution). After these samples are drawn, the GLMM is reduced to a full data scenario with no missing data. PROC BGLIMM then proceeds to update $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, **G**, and **R** sequentially.

Sampling Methods

Sampling methods that PROC BGLIMM uses include the conjugate sampler, direct sampler, Gamerman algorithm (a variation of the Metropolis-Hastings algorithm that is tailored to generalized linear models), and No-U-Turn Sampler (NUTS, a self-tuning variation of the Hamiltonian Monte Carlo (HMC) method).

The conjugate sampler is used in normal models and in sampling variance and covariance parameters when conjugate priors are specified. The Gamerman algorithm is used for both the fixed-effects and random-effects parameters in nonnormal models. Missing values are sampled using direct sampling methods. The NUTS algorithm is used for covariance parameters when conjugacy is not available.

PRIOR DISTRIBUTIONS

PROC BGLIMM sets default prior distributions for all parameters in the model. You can use options in the MODEL, RANDOM, and REPEATED statements to modify prior distributions for the fixed effects, the scale parameters (in applicable likelihood functions), the G-side matrix, and the R-side matrix. The prior distribution for random effects is either univariate normal or multivariate normal, and that cannot be changed.

For fixed-effects parameters, the default prior is a flat prior, and you can use the CPRIOR= option in the MODEL statement to specify a normal prior (for example, **`cprior=normal(var=1e4)`**) or use a data set (which should contain hyperparameter mean and covariance values) to specify a multivariate normal prior (for example, **`cprior=normal(input=MyPrior)`**, where **MyPrior** is the name of the SAS data set).

The TYPE= option in the RANDOM statement controls the G-side covariance type, and the COVPRIOR= option specifies the prior distributions for the parameters in the **G** matrix. This option is applicable to the UN, UN(1), VC, and TOEP covariance types. Parameters in other G-side covariance matrix are given a flat prior distribution, and that cannot be changed in this release.

For the UN, UN(1), VC, and TOEP G-side covariance types, you can specify an inverse Wishart prior (with diagonal scale matrix), a scaled inverse Wishart prior (with diagonal scale matrix), an inverse gamma prior, a uniform prior, a half-Cauchy prior, and a half-normal prior. Among the scalar prior distributions, the uniform prior is applicable to the standard deviation, and the other priors are applicable to the variance parameters.

You use the SCALEPRIOR= option in the MODEL statement to specify a prior on the scale parameters in four distributions (likelihood functions): normal, negative binomial, gamma, and inverse gamma. You can choose an inverse gamma prior, a gamma prior, or an improper prior ($\pi(\phi) \propto 1/\phi$)), although only the inverse gamma is applicable to the normal likelihood function with the identity link.

You use the COVPRIOR= option in the REPEATED statement to specify a prior distribution on the R-side variance-covariance matrix. You can choose an inverse Wishart prior (with diagonal scale matrix) or an inverse gamma prior. When you specify the COVPRIOR= option, this prior overrides the prior that you specify in the SCALEPRIOR= option in the MODEL statement for normal data.

EXAMPLES

Example 1: Logistic Regression with Random Intercepts

This example demonstrates how you can use PROC BGLIMM to fit a mixed model to binomial data.

Researchers investigated the performance of two medical procedures in a multicenter study. They randomly selected 15 centers for inclusion. One of the study goals was to compare the occurrence of side effects from the procedures. In each center, n_{A} patients were randomly selected and assigned to treatment group A, and n_{B} patients were randomly assigned to treatment group B. The following DATA step creates the data set, **MultiCenter**, for the analysis:

```
data MultiCenter;
   input Center Group$ N SideEffect @@;
   datalines;
 1  A  32  14   1  B  33  18
 2  A  30   4   2  B  28   8
 3  A  23  14   3  B  24   9
 4  A  22   7   4  B  22  10
 5  A  20   6   5  B  21  12
 6  A  19   1   6  B  20   3
 7  A  17   2   7  B  17   6
 8  A  16   7   8  B  15   9
 9  A  13   1   9  B  14   5
10  A  13   3  10  B  13   1
11  A  11   1  11  B  12   2
12  A  10   1  12  B   9   0
13  A   9   2  13  B   9   6
14  A   8   1  14  B   8   1
15  A   7   1  15  B   8   0
;
```

The variable **Group** identifies the two medical procedures, **N** is the number of patients who received a given procedure at a particular center, and **SideEffect** is the number of patients who reported side effects.

If $\mathrm{y}_{i\mathrm{A}}$ and $\mathrm{y}_{i\mathrm{B}}$ denote the number of patients at center i who reported side effects for procedures A and B, respectively, then for a given center these are independent binomial random variables. To model the probability of having side effects from the two procedures, $p_{i\mathrm{A}}$ and $p_{i\mathrm{B}}$, you need to account for the fixed group effect and the random selection of centers. One possibility is to assume a model that relates group and center effects linearly to the logit of the probabilities:

$$\log\left\{\frac{p_{i\mathrm{A}}}{1-p_{i\mathrm{A}}}\right\} = \beta_{\mathrm{A}} + \gamma_i$$
$$\log\left\{\frac{p_{i\mathrm{B}}}{1-p_{i\mathrm{B}}}\right\} = \beta_{\mathrm{B}} + \gamma_i$$

In this model, β_{A} and β_{B} are fixed effects, and $\beta_{\mathrm{A}} - \beta_{\mathrm{B}}$ measures the difference in the logits of experiencing side effects; the γ_i are independent random variables that result from the random selection of centers. Observations from the same center receive the same adjustment, and these adjustments vary randomly from center to center, with variance $\mathrm{Var}[\gamma_i] = \sigma_c^2$.

Because $p_{i\mathrm{A}}$ is the conditional mean of the sample proportion, $\mathrm{E}[\mathrm{y}_{i\mathrm{A}}/n_{i\mathrm{A}}|\gamma_i] = p_{i\mathrm{A}}$, you can model the sample proportions as binomial ratios in a generalized linear mixed model. The following statements perform this analysis under the assumption of normally distributed center effects with equal variance and a logit link function:

```
ods graphics on;
proc bglimm data=MultiCenter nmc=10000 thin=2 seed=976352
   plots=all;
   class Center Group;
   model SideEffect/N = Group / noint;
   random int / subject = Center;
run;
```

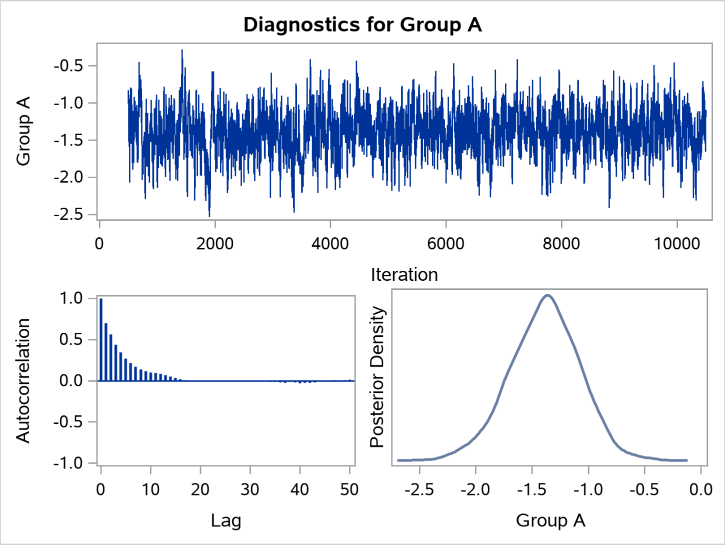
PROC BGLIMM produces posterior estimates (in the “Posterior Summaries and Intervals” table in Figure 1) for the fixed coefficients ($\boldsymbol{\beta}$) and the variance of the random center intercepts (σ_c^2). Because of the fixed-effects parameterization that is used here, the “Group A” effect is an estimate of β_{A} (–1.39), and the “Group B” effect is an estimate of β_{B} (–0.88). The two estimates show that there is a difference between the two groups. By default, posterior summary statistics of random-effects parameters are not displayed. You can display them by using the MONITOR option in the RANDOM statement.

Figure 1 Posterior Summaries and Intervals

Posterior Summaries and Intervals					
			Standard	95%	
Parameter	N	Mean	Deviation	HPD Interval	
Group A	5000	-1.3895	0.3102	-2.0071	-0.7956
Group B	5000	-0.8839	0.2968	-1.4819	-0.3186
Random Var	5000	0.9184	0.4198	0.3024	1.7515

PROC BGLIMM also produces trace plots, autocorrelation plots, and density plots of model parameters, as shown in Figure 2.

Figure 2 PROC BGLIMM Diagnostic Plots



You can use the autocall macro %TADPLOT to regenerate the same diagnostic plots for any selected parameters.

Use the ESTIMATE statement as follows to compute the log of odds ratios between the two treatment groups, A and B:

```
proc bglimm data=MultiCenter nmc=10000 thin=2 seed=976352
   outpost=CenterOut;
   class Center Group;
   model SideEffect/N = Group / noint;
   random int / subject=Center monitor;
   estimate "log OR" group 1 -1;
run;
```

The ESTIMATE statement computes $\beta_{\mathrm{A}} - \beta_{\mathrm{B}}$ for every posterior sample, and the transformed variable values are saved to the OUTPOST= data set under the variable name **Log_or**.

The table in Figure 3 shows that the posterior mean of the log of odds ratio is around –0.5, with the 95% HPD interval all negative. This indicates that patients who undergo procedure A have a lower chance of developing side effects than patients who undergo procedure B.

Figure 3 Estimated Differences in the Logits

The BGLIMM Procedure

Results from ESTIMATE Statements				
		Standard	95%	
Label	Mean	Deviation	HPD Interval	
log OR	-0.5056	0.2087	-0.9292	-0.1102

The ESTIMATE statement does not compute the difference in probabilities of side effects directly. You compute this difference by using a DATA step, where **CenterOut** is the saved OUTPOST= data set from a previous PROC BGLIMM run:

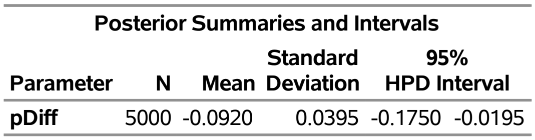
```
data prob;
   set CenterOut;
   pDiff = logistic(group_a) - logistic(group_b);
run;
```

You can use the “%SUMINT” autocall macro to compute the posterior summary statistics of **pDiff**:

```
%sumint(data=prob, var=pDiff)
```

The results are shown in Figure 4. As you can see, there is a significant difference in probabilities of side effects between the two groups.

Figure 4 Posterior Summary Statistics



Posterior Summaries and Intervals

			Standard	95%	
Parameter	**N**	**Mean**	**Deviation**	**HPD Interval**	
pDiff	5000	-0.0920	0.0395	-0.1750	-0.0195

.

Example 2: Multilevel Clinical Trial

This example illustrates how to use PROC BGLIMM to analyze hierarchical data that have nested clusters at multiple levels. It also demonstrates how you can use the deviance information criterion (DIC) to evaluate the fit of a model.

Brown and Prescott (1999) discussed a clinical trial for hypertension in which 288 patients at 29 centers were randomized to receive one of three hypertension treatments: a new drug, Carvedilol; and two standard drugs, Nifedipine and Atenolol. Patients were followed up four times, once every other week for four visits. A baseline diastolic blood pressure (DBP) was recorded before the treatment, and the DBP was recorded again at each of the four follow-up visits. One goal of this study was to assess the effect of the three treatments on DBP over the follow-up period.

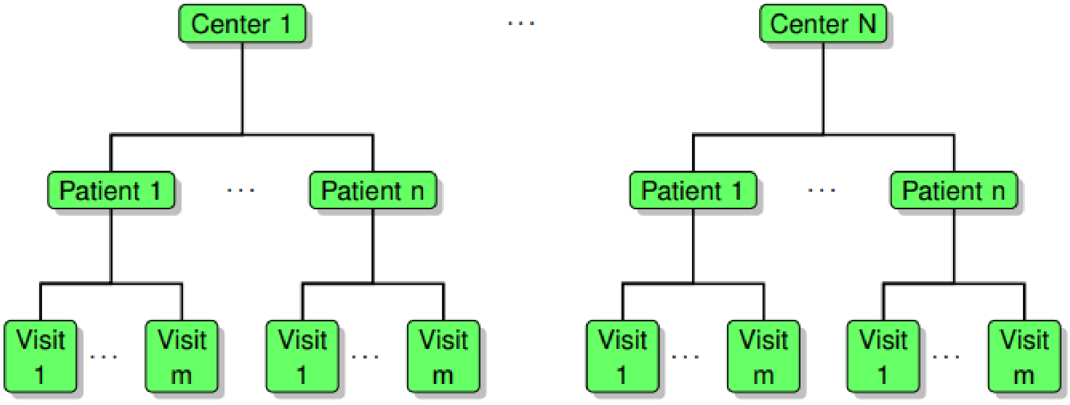
The following statements show part of the data set:

```
data DBP;
   input Patient Visit Center Treat$ DBP DBP1;
   datalines;
79 3 1 Carvedil 96 100
79 4 1 Carvedil 108 100
80 3 1 Nifedipi 82 100
80 4 1 Nifedipi 92 100
80 5 1 Nifedipi 90 100
80 6 1 Nifedipi 100 100
81 3 1 Atenolol 86 100

   ... more lines ...

237 5 41 Atenolol 80 104
237 6 41 Atenolol 90 104
238 3 41 Nifedipi 88 112
238 4 41 Nifedipi 100 112
;
```

In the INPUT statement, the variable **Patient** identifies patients; the variable **Visit** records the visit time 3, 4, 5, or 6 after randomization; the variable **Center** represents the center that each patient visited; the variable **Treat** indicates which treatment group (Carvedilol, Nifedipine, or Atenolol) each patient was assigned to; **DBP** is the diastolic blood pressure (in mmHg) measured at each follow-up visit; and **DBP1** is the baseline diastolic blood pressure measured before randomization.

As shown in Figure 5, this study has a three-level structure, where the **Visit** is the level-1 unit at the bottom of the hierarchy, the **Patient** is the level-2 unit, and the **Center** is the level-3 unit at the top. Visits are nested within patients, which are nested within centers. The units at levels higher than level 1 are sometimes called clusters. Visit time is a level-1 covariate. Baseline DBP and treatment vary only from patient to patient and are thus level-2 covariates. No level-3 covariates are measured on the centers.

Figure 5 Three-Level Structure

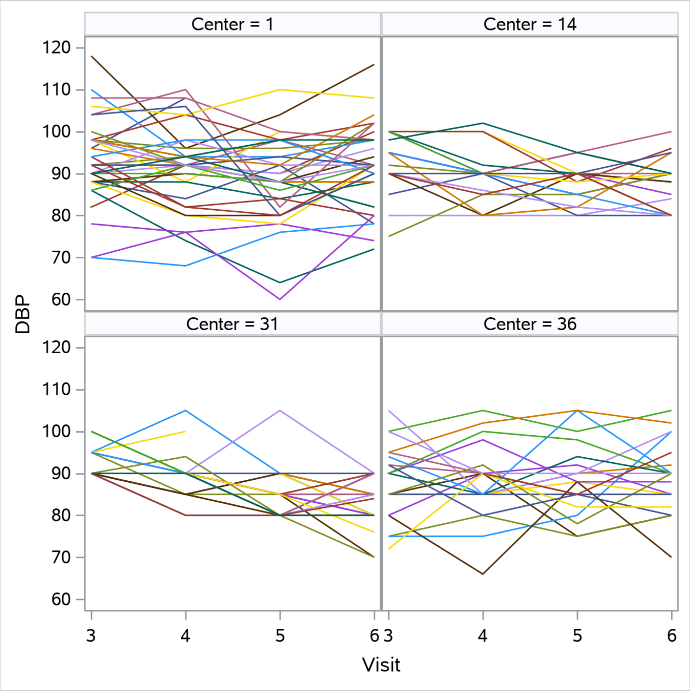


Patients at the same center tend to be more similar to each other than they are to patients from another center. The reason for within-center similarity could be the closeness of patients' residences or the shared medical practice at the center. Furthermore, repeated DBP measurements of the same patient are closer to each other than they are to measurements of a different patient.

The within-cluster dependence makes ordinary regression modeling inappropriate, but you can use multilevel models to accommodate such dependence. However, the cluster correlation is more than just a nuisance. The hierarchical design provides rich information about how the processes operate at different levels. Multilevel models enable you to disentangle such information by including covariates at different levels and assigning unexplained variability to different levels. For example, a three-level model enables you to estimate effects of covariates at the visit, patient, and center levels in the multicenter study. Moreover, you can include random effects to address the variability that is not explained by those covariates. These random effects are specified at levels that are defined by nested clusters.

Figure 6 shows a spaghetti plot of DBP against the visit time for the patients at four centers. The plot shows that the DBP trends vary significantly from patient to patient. If you picture the trend of DBP as a linear function of visit time, you can see considerable variability in the intercepts within each center.

Figure 6 Spaghetti Plot of Four Centers



Consider constructing a three-level model in the following stages:

1. The level-1 model for visit i of patient j at center k is a linear regression on visit time,

$$\text{DBP}_{ijk} = \alpha_0 + \alpha_1 \text{Visit}_{ijk} + e_{ijk}$$

2. Assume that the intercept α_0 varies among patients according to the level-2 model,

$$\alpha_0 = \delta_0 + \delta_1 \text{Baseline}_{jk} + \delta_2 \text{Treat}_{jk} + \gamma_{0,jk}^{\text{Patient}} \text{Patient}_{jk}$$

where $\gamma_{0,jk}^{\text{Patient}}$ is the patient-level random intercept.

3. Express the variability among the centers in the level-3 model,

$$\delta_0 = \lambda_0 + \gamma_{0,k}^{\text{Center}} \text{Center}_k$$

where $\gamma_{0,k}^{\text{Center}}$ is the center-level random intercept.

Substituting the level-3 model into the level-2 model and then substituting the level-2 model into the level-1 model yields

$$\begin{aligned} \text{DBP}_{ijk} &= \lambda_0 + \alpha_1 \text{Visit}_{ijk} + \delta_1 \text{Baseline}_{jk} + \delta_2 \text{Treat}_{jk} \\ &\quad + \gamma_{0,k}^{\text{Center}} \text{Center}_k \\ &\quad + \gamma_{0,jk}^{\text{Patient}} \text{Patient}_{jk} \\ &\quad + e_{ijk} \end{aligned}$$

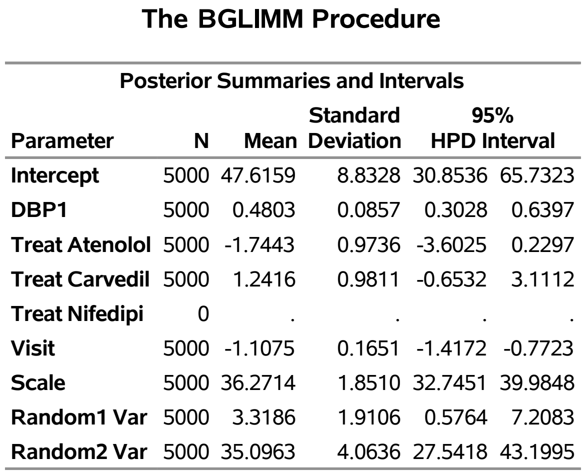
You can fit this three-level model by using the following PROC BGLIMM code:

```
proc bglimm data=DBP seed=98876 nmc=10000 thin=2 dic;
   class Patient Center Treat;
   model DBP = DBP1 Treat Visit ;
   random intercept / subject = Center;
   random intercept / subject = Patient(center);
   estimate 'Carvedil vs. Atenolol' Treat -1 1  0;
   estimate 'Carvedil vs. Nifedipi' Treat 0  1 -1;
run;
```

The two RANDOM statements specify two random intercepts with different clustering, and the second RANDOM statement has a nested subject. The two ESTIMATE statements compare the effect of the new drug, Carvedilol, with the effects of the two standard treatments, Nifedipine and Atenolol.

The “Posterior Summaries and Intervals” table in Figure 7 lists the summary statistics (posterior means, standard deviations, and HPD intervals) for each parameter, the fixed coefficients ($\boldsymbol{\beta}$), the scale parameter (σ^2), the variance at the center level (labeled “Random1 Var” because it is the first RANDOM statement), and the variance at the patient level (labeled “Random2 Var”). If you control for other covariates, the DBP decreases 1.11 mmHg (see the posterior mean for **Visit**) at each successive visit on average, and every increase of 1 mmHg in baseline DBP leads to an increase of 0.48 mmHg (see the posterior mean for **DBP1**) in posttreatment DBP. You can see that the “Treat Atenolol” effect (versus the effect of the reference group, “Treat Nifedipi,” which is fixed at 0) is –1.74 and the “Treat Carvedil” effect is 1.24.

Figure 7 Posterior Summaries and Intervals



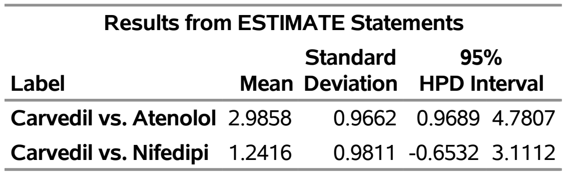
The BGLIMM Procedure

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Intercept	5000	47.6159	8.8328	30.8536	65.7323
DBP1	5000	0.4803	0.0857	0.3028	0.6397
Treat Atenolol	5000	-1.7443	0.9736	-3.6025	0.2297
Treat Carvedil	5000	1.2416	0.9811	-0.6532	3.1112
Treat Nifedipi	0	.	.	.	.
Visit	5000	-1.1075	0.1651	-1.4172	-0.7723
Scale	5000	36.2714	1.8510	32.7451	39.9848
Random1 Var	5000	3.3186	1.9106	0.5764	7.2083
Random2 Var	5000	35.0963	4.0636	27.5418	43.1995

Figure 8 shows the results of comparing the effects of the new drug (Carvedilol) with the effects of the two standard treatments (Nifedipine and Atenolol). You can see that the DBP of a patient who receives Carvedilol is 3 mmHg higher, on average, than the DBP of a patient who receives Atenolol and that the 95% HPD interval does not include 0. The DBP of a patient who receives Carvedilol is 1.2 mmHg higher, on average, than the DBP of a patient who receives Nifedipine, but the 95% HPD interval includes 0.

Figure 8 Estimated Differences in Treatments



Results from ESTIMATE Statements

Label	Mean	Standard Deviation	95% HPD Interval	
Carvedil vs. Atenolol	2.9858	0.9662	0.9689	4.7807
Carvedil vs. Nifedipi	1.2416	0.9811	-0.6532	3.1112

You can compute the conditional correlation between DBP measurements of two different patients at the same center and the conditional correlation between DBP measurements of the same patient at two visits. You can do this by using the posterior means of the scale parameter (σ^2), the variance at the center level, and the variance at the patient level, as follows:

$$\text{Corr}(\text{DBP}_{ijk}, \text{DBP}_{i'j'k}) = \frac{3}{3+35+36} = 0.04$$

$$\text{Corr}(\text{DBP}_{ijk}, \text{DBP}_{i'jk}) = \frac{3+35}{3+35+36} = 0.51$$

That is, of the variability in DBP that is not explained by the covariates, 4$\%$ is caused by unobserved center-specific attributes and 51$\%$ is caused by unobserved patient-specific attributes. Another way to interpret this is that DBP measurements for the same patient are much more similar to each other than are DBP measurements for different patients at the same center, as the spaghetti plot in Figure 6 indicates.

The analysis so far has revealed that visit time has a very strong negative effect on DBP. That is, the average patient's blood pressure decreases over the course of the study, regardless of treatment or center. Is this reduction rate the same for all centers? This is a question about the interaction between **Center** and **Visit**. The three-level model that was previously posited assumes that only the intercept varies among centers, but now you want to know whether the slope for time also varies among centers:

$$\begin{aligned}\text{DBP}_{ijk} &= \lambda_0 + \alpha_1 \text{Visit}_{ijk} + \delta_1 \text{Baseline}_{jk} + \delta_2 \text{Treat}_{jk} \\ &\quad + \gamma_{0,k}^{\text{Center}} \text{Center}_k + \gamma_{1,k}^{\text{Center}} \text{Visit}_{ijk} \text{Center}_k \\ &\quad + \gamma_{0,jk}^{\text{Patient}} \text{Patient}_{jk} \\ &\quad + e_{ijk}\end{aligned}$$

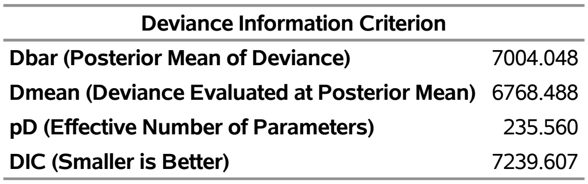
where $\gamma_{1,k}^{\text{Center}}$ is the center-level random slope for visit time.

You can fit this modified three-level model with both random intercept and slope at the center level by using the following code:

```
proc bglimm data=DBP seed=98876 nmc=10000 thin=2 dic;
   class Patient Center Treat;
   model DBP = DBP1 Treat Visit ;
   random intercept Visit / subject = Center type=un;
   random intercept / subject = Patient (center);
run;
```

The TYPE=UN option in the first RANDOM statement requests an unstructured covariance structure for center-level random effects. The DIC option in the PROC BGLIMM statement calculates the deviation information criterion, which results in a DIC value of 7239.6 (Figure 9). The random-intercept-only model, in contrast, has a larger DIC value of 7253.264 (Figure 10). The added parameters in the random-intercept-and-slope model provide a better fit for the model.

Figure 9 DIC Values from Random-Intercept-and-Slope Model



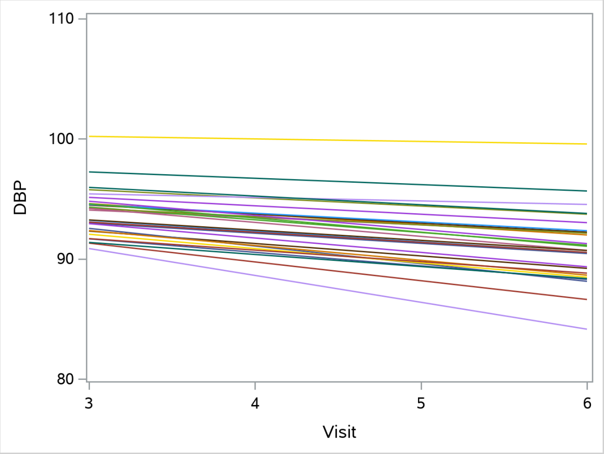
Deviance Information Criterion	
Dbar (Posterior Mean of Deviance)	7004.048
Dmean (Deviance Evaluated at Posterior Mean)	6768.488
pD (Effective Number of Parameters)	235.560
DIC (Smaller is Better)	7239.607

Figure 10 DIC Values from Random-Intercept-Only Model

Deviance Information Criterion	
Dbar (Posterior Mean of Deviance)	7024.194
Dmean (Deviance Evaluated at Posterior Mean)	6795.124
pD (Effective Number of Parameters)	229.070
DIC (Smaller is Better)	7253.264

Figure 11 plots the predicted DBP over time for each center. You can see that both intercept and slope vary significantly across centers.

Figure 11 Predicted DBP over Time across Centers



Example 3: Repeated Measurements with Heterogeneous Variance

Heterogeneity of variances occurs in many situations. A main motivation for modeling heterogeneous variances is to appropriately down-weight portions of the data that are highly variable and extract more information from portions of the data that are less variable.

As discussed in Littell et al. (2006), heterogeneous models fall into two categories: within-subject heterogeneity of the covariance parameters and between-subject heterogeneity of the covariance parameters. Within-subject heterogeneity occurs across data from the same individual. An example is the variances that change with time in a longitudinal or repeated-measures setting. Between-subject heterogeneity occurs when different groups of subjects display different variance patterns but are homogeneous within groups or when the variance components that correspond to random effects are unequal. Heterogeneous variances can be incorporated into the analysis if you specify various variance or covariance structures.

This example illustrates the two types of heterogeneity in the context of repeated-measures data. The data come from a two-treatment, randomized double-blind clinical trial for patients with rheumatoid arthritis; see Patel (1991). Sixty-seven patients enrolled in the trial. A baseline grip strength (in mmHg) was measured at the start of the trial. All patients were followed up three times, and a grip strength measurement was taken at each follow-up visit. The distribution of grip strength among males was expected to have a higher mean value than that among females.

The following DATA step reads the data set, **GripData**:

```
data GripData;
   input Subject Baseline Treat Gender $ Time T Grip;
   datalines;
26  175  1  M  1  1  161
26  175  1  M  2  2  210
26  175  1  M  3  3  230
27  165  1  M  1  1  215
27  165  1  M  2  2  245
27  165  1  M  3  3  265
```

```
   ... more lines ...

71  104  2  F  1  1  107
71  104  2  F  2  2  .
71  104  2  F  3  3  .
72   60  2  F  1  1   60
72   60  2  F  2  2   55
72   60  2  F  3  3   58
;
```

Some response data are missing; that is, some patients were measured on fewer than three occasions. By default, PROC BGLIMM treats missing response values as parameters and includes the sampling of the missing data as part of the simulation. The procedure discards all observations that have missing covariates. This is equivalent to assuming that the missing values are missing at random (MAR).

Initial Model

A reasonable initial model for most data should involve fairly general specifications for both the mean and the variance-covariance structure (as recommended by Littell et al. 2006). This initial model includes a **Baseline** covariate and three-way interactions of the class variables **Gender**, **Treat**, and **Time**. To allow general within-subject heterogeneity, the unstructured covariance is used in the R-side matrix. An advantage of considering this most general model is that you can inspect the estimates of the covariance matrix for heterogeneous patterns in both the variances and correlations.

You can fit the initial model by using the following code:

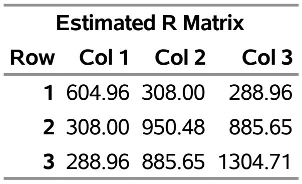
```
proc bglimm data=GripData seed=475193 dic;
   class Subject Treat Gender Time;
   model Grip = Gender*Treat*Time Baseline / noint;
   repeated Time / sub=Subject type=un r rcorr;
   run;
```

The MODEL statement specifies that the response variable is **Grip** and that the fixed effects contain 12 cell means involving **Gender**, **Treat**, and **Time** (2 treatments by 2 genders by 3 visits). The crossed effects (interactions) are specified by joining the three classification variables with asterisks as a simple way to obtain the 12 main cell means; you could also use the vertical bar operator (|) as shorthand for all main effects and interactions, which should produce an equivalent model with different interpretations for some parameters. In addition, the mean model includes a baseline covariate.

The REPEATED statement specifies that the repeated measurements be taken over the **Time** variable. The repeated effect is required in a REPEATED statement, and it must be specified as a CLASS variable. The repeated measurements are grouped according to **Subject** (the SUB= variable), and the covariance type is specified as unstructured via the TYPE=UN option. The R and RCORR options produce printouts of the estimated covariance matrix of $\mathbf{R}$ and its corresponding correlation form.

Figure 12 shows the estimated **R** covariance in the 3×3 matrix format and its correlation form. The variances appear to increase over time. And there is no obvious pattern in the correlation structures—an indication that the fully unstructured type might be necessary.

Figure 12 Estimated Covariance and Correlation Matrices of $\mathbf{R}$



Estimated R Matrix

Row	Col 1	Col 2	Col 3
1	604.96	308.00	288.96
2	308.00	950.48	885.65
3	288.96	885.65	1304.71

Figure 12 *continued*

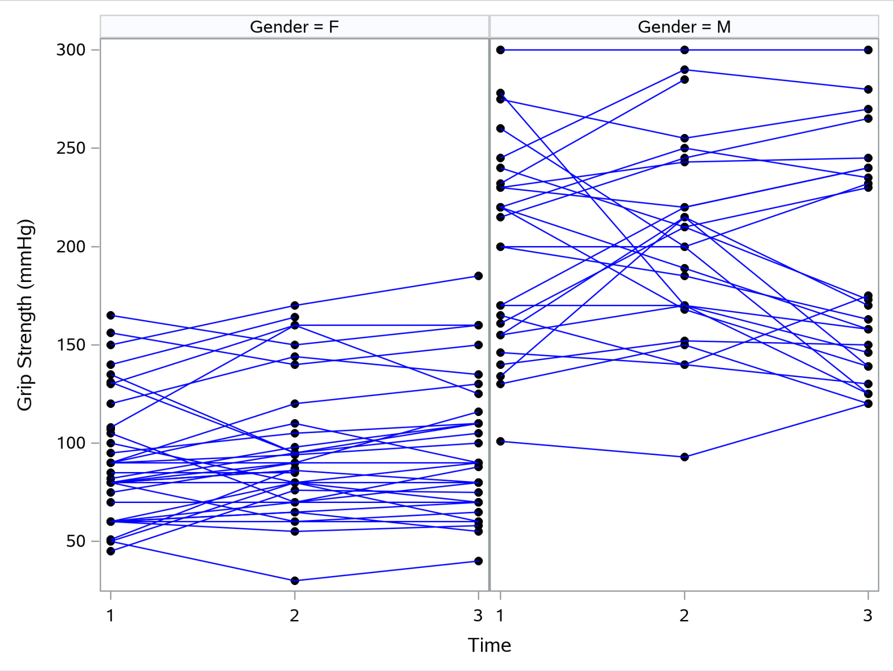
Estimated R Correlation Matrix

Row	Col 1	Col 2	Col 3
1	1.0000	0.4062	0.3252
2	0.4062	1.0000	0.7953
3	0.3252	0.7953	1.0000

Between-Subject Heterogeneity

There is, however, a considerate amount of between-subject heterogeneity in the data. To show this, Figure 13 plots side-by-side profiles of grip strength by time for female and male patients. Males tend to have a stronger grip and higher levels of variability across visits.

Figure 13 Grip Strength Plot by Gender



To account for distinct covariance structures of the two gender groups, you can fit the model by adding the option GROUP=GENDER to the REPEATED statement:

```
proc bglimm data=GripData seed=475193;
   class Subject Treat Gender Time;
   model Grip = Gender*Treat*Time Baseline / noint;
   repeated Time / subject=Subject type=un group=Gender r;
   run;
```

Figure 14 displays the estimated covariance matrices (the first three rows are for female patients and the last three rows are for male patients). They indicate three systematic between-gender differences: (1) male patients have higher variances across the board; (2) variances for female patients decrease over time, but the trend is reversed for males; (3) the correlation patterns are not the same if you compare the off-diagonal terms between the two gender groups.

Figure 14 Estimated Covariance of $\mathbf{R}$ for Both Genders

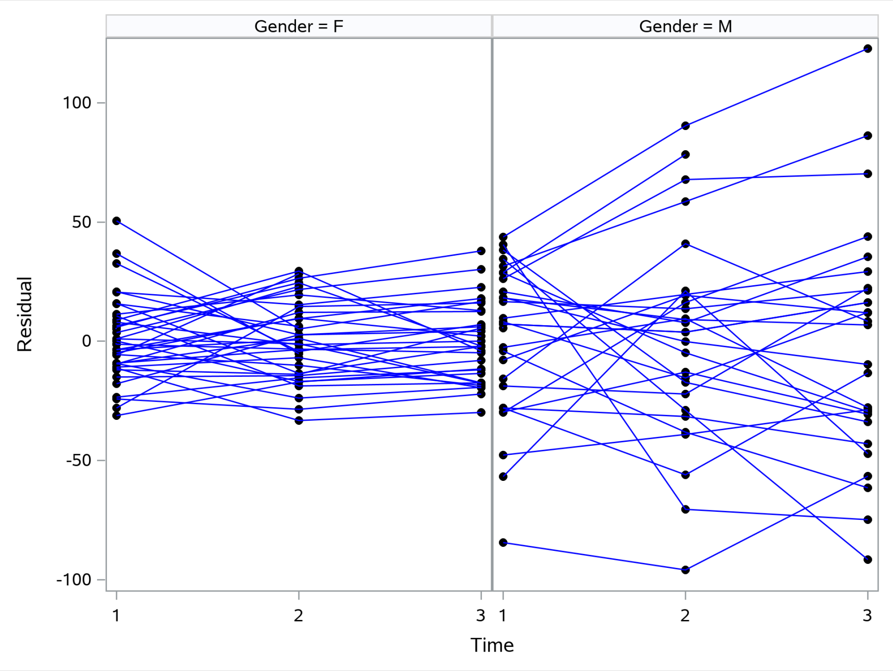
The BGLIMM Procedure

Estimated R Matrix

Group	Row	Col 1	Col 2	Col 3
Gender F	**1**	300.08	77.2769	95.2165
Gender F	**2**	77.2769	267.23	195.20
Gender F	**3**	95.2165	195.20	257.48
Gender M	**1**	960.37	591.43	528.93
Gender M	**2**	591.43	1773.63	1710.94
Gender M	**3**	528.93	1710.94	2504.11

You might notice that the increase and decrease of variances over time are not apparent in the data plot in Figure 13. This is because most of the overall variabilities in the data are explained by the baseline covariates, and the remaining variabilities are modeled by the **R** matrix. Figure 15 shows that the residuals and the variance trends are in closer agreement with the estimates.

Figure 15 Residuals Plot by Gender



Alternative Approach

As an alternative to the previous model, you can account for both between- and within-subject heterogeneity by using a random-effects model. Consider a random-intercept model with heterogeneous residual variance for the two genders, which you specify using the following statements:

```
proc bglimm data=GripData seed=475193 nmc=20000 thin=4;
   class Subject Treat Gender Time;
   model Grip = Gender*Treat*Time Baseline / noint;
   random int / sub=Subject group=Gender covprior=uniform;
   repeated Time / sub=Subject type=un group=Gender r rcorr covprior=iw(scale=500);
   run;
```

The RANDOM statement is added to account for the between-subject heterogeneity; the GROUP=GENDER option indicates that patients have different variances between genders but are homogeneous within each gender. The COVPRIOR=UNIFORM option in the RANDOM statement specifies a noninformative prior for the variance parameter to downplay the role of a relatively informative prior on the posterior distribution. The COVPRIOR=IW(SCALE=500) option in the REPEATED statement changes to use a much larger scale hyperparameter (from the default 4 to 500) for the same purpose.

The random-effects model shifts some of the variability from the **R** matrix to the G side, resulting in smaller residual variance estimates.

COMPARISON WITH PROC MIXED, PROC GLIMMIX, AND PROC MCMC

PROC MIXED and PROC GLIMMIX provide classical frequentist statistics solutions to the linear and generalized linear mixed-effects models, respectively. Frequentist estimation methods rely on maximizing the marginal likelihood function, and inferences are often based on asymptotic theorems. In linear model scenarios, PROC MIXED and PROC BGLIMM produce estimates that are nearly identical when noninformative prior distributions are used in PROC BGLIMM. PROC GLIMMIX can produce estimates that are quite different from those of PROC BGLIMM, in situations where linearization methods (such as pseudo-likelihood estimation) are used in PROC GLIMMIX.

PROC MCMC is a general-purpose, simulation-based Bayesian procedure that provides flexibility in model specification but requires more programming by the user. You can use PROC MCMC to fit GLMMs, although mixing can sometimes be less efficient because of the general sampling (and not model-specific) algorithms that PROC MCMC uses (Chen, Brown, and Stokes 2016). PROC MCMC also lacks some conveniences; for example, it does not support a CLASS statement to handle categorical variables automatically.

Certain features have yet to be implemented in PROC BGLIMM, and you need to use PROC MCMC instead. For example, if you want to work with more general prior distributions (on fixed effects, G-sided variance terms, and so on), or specify random effects with nonnormal prior distributions or nested prior distributions, or if you want to work with missing not at random data, and so on, then you need to use PROC MCMC.

SUMMARY

PROC BGLIMM is a Bayesian procedure that is designed specifically for fitting generalized linear mixed models by using Markov chain Monte Carlo methods. The procedure adopts familiar SAS syntax in specifying GLMMs, and a key enhancement over the existing MCMC procedure is its simplicity in specifying a large class of GLMMs. PROC BGLIMM uses efficient sampling algorithms that are parallelized for performance, resulting in good mixing and faster computation. PROC BGLIMM also provides functionality for handling missing data, nested multilevel models, and repeated-measures data. Additional features will be incorporated in future releases.

REFERENCES

Brown, H., and Prescott, R. (1999). *Applied Mixed Models in Medicine*. New York: John Wiley & Sons.

Chen, F., Brown, G., and Stokes, M. (2016). “Fitting Your Favorite Mixed Models with PROC MCMC.” In *Proceedings of the SAS Global Forum 2016 Conference*. Cary, NC: SAS Institute Inc. `https://support.sas.com/resources/papers/proceedings16/SAS5601-2016.pdf`.

Gamerman, D. (1997). “Sampling from the Posterior Distribution in Generalized Linear Models.” *Statistics and Computing* 7:57–68.

Hoffman, M. D., and Gelman, A. (2014). “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo.” *Journal of Machine Learning Research* 15:1351–1381.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006). *SAS for Mixed Models*. 2nd ed. Cary, NC: SAS Institute Inc.

Neal, R. M. (2011). “MCMC Using Hamiltonian Dynamics.” In *Handbook of Markov Chain Monte Carlo*, edited by S. Brooks, A. Gelman, G. L. Jones, and X.-L. Meng, 113–161. Boca Raton, FL: CRC Press.

RECOMMENDED READING

PROC BGLIMM requires SAS® 9.4M6. Complete documentation of the BGLIMM procedure can be found on the web at http://support.sas.com/documentation/onlinedoc/stat/151/bglimm.pdf.

You can find additional coding examples at http://support.sas.com/rnd/app/examples/STATexamples.html.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Amy Shi
SAS Institute Inc.
SAS Campus Drive, Cary, NC 27513
amy.shi@sas.com

SAS® GLOBAL FORUM 2021

Paper 1146-2021

Bayesian Analysis of GLMMs Using PROC BGLIMM

Walter W. Stroup, University of Nebraska-Lincoln

ABSTRACT

Over the past two decades, generalized linear mixed models (GLMMs) have become standard tools for statistical analysis. Since its introduction, PROC GLIMMIX has been SAS®/STAT's primary GLMM procedure. The most recent edition of *SAS for Mixed Models* includes three chapters on using GLIMMIX for GLMMs.

PROC GLIMMIX is an excellent frequentist tool. However, Bayesian approaches are becoming increasingly important. Many academic journals prefer - some even require - Bayesian analysis. Even when not required, Bayesian methods allow you to use what you know prior to, or in the early stages of, an investigation.

PROC BGLIMM is a new SAS/STAT procedure that makes Bayesian implementation of GLMMs relatively easy. BGLIMM uses syntax similar to PROC GLIMMIX, but there are some differences. This tutorial presents what you need to know to get started using PROC BGLIMM. We use GLMM examples from *SAS for Mixed Models*, but with a Bayesian twist.

INTRODUCTION

Mixed models are important tools for analyzing data from many types of studies, including longitudinal or repeated measures, multi-level or split-plot experiments, blocked designs with incomplete blocks or missing data and multi-location studies. Linear mixed models (LMMs) accommodate response variables assumed to follow a normal (hereafter referred to as a Gaussian) distribution. Generalized linear mixed models (GLMMs) extend mixed model theory and methods to accommodate non-Gaussian responses such as categorical or count data. Because the LMM is a special case of the GLMM – a GLMM with Gaussian data – the acronym GLMM is used for the rest of this paper. In the SAS® system, PROC GLIMMIX is the preeminent mixed model procedure, allowing users to work with both LMMs and GLMMs. PROC GLIMMIX uses a **frequentist** approach to estimation and inference. The next section, "GLMM Basics" gives an overview of what this entails, but the basic idea is that frequentist statistics focus on obtaining estimates and standard errors that are used to construct test statistics for significance testing or to construct confidence intervals.

Bayesian statistics provide an alternative approach. Bayesian statistics focus on incorporating prior information to obtain posterior distributions of the statistics of interest. Posterior distributions combine what you know in advance – the prior – with the data you observe. Bayesian statistics are becoming increasingly important for data analysis. One reason is that many academic journals now discourage classical significance testing in favor of Bayesian analysis. Some journals have even banned significance tests, in effect requiring Bayesian analysis. The second important reason is that the statistics obtained from frequentist methods are essentially equivalent to Bayesian analysis with a "non-informative" prior. In other words, frequentist methods implicitly assume that you know nothing until you analyze the data. In reality, this is rarely, if ever, true (if you really know nothing before analyzing the data, you probably should not be doing the study!). Many data sets come from studies that may be part of a series – phases of clinical trials leading to the licensing of a pharmaceutical product – or similar experiments by graduate students with the same advisor.

PROC MCMC is the SAS system's original and all-purpose Bayesian procedure. Although you can use PROC MCMC for mixed models, the syntax is similar to PROC NLMIXED, which can be inconvenient for less-than-simple models or models with CLASS variables. PROC BGLIMM was developed to allow you to use syntax similar to PROC GLIMMIX, making Bayesian analysis more accessible for GLMMs.

This tutorial introduces PROC BGLIMM using three examples from *SAS for Mixed Models: Introduction and Basic Applications* (Stroup, et al. 2018).

- A multi-clinic trial with binomial data from Chapter 11 of *SAS for Mixed Models*.
- A multi-level (split-plot) experiment with count data from Chapter 13.
- Repeated measures (longitudinal) data from Chapter 8.

The next section called "GLMM Basics," covers the GLMM setting, essential definitions and terminology, and an overview and comparison of PROC GLIMMIX and PROC BGLIMM estimation and inference.

GLMM BASICS

The classical format for a statistical model is the equation

Response variable = systematic component + random component

Think of the systematic component as the fixed effects, e.g., $\mu + \tau_i$ in ANOVA-type models with treatment effects or $\beta_0 + \beta_i X$ in linear regression models. In mixed models, the random component is really two distinct components, random model effects such as block effects, and residual effects, such as those that account for serial covariance in a longitudinal mixed model. The classical response = fixed + random format works well if you can assume that the data follow a Gaussian distribution, but in many cases this is not true. Table 1 shows common types of response variables in the left-hand column and types of model effects across the top row.

Response Variable	Fixed		Random	
	Categorical (CLASS) $\mu + \tau_i$	Continuous $\beta_0 + \beta_i X$	Model effect	Residual / Covariance structure
Gaussian				"R-side"
Categorical binomial multinomial				"G-side"
Continuous proportion beta				
Count Poisson negative binomial				
time-to-event				
etc...				

Table 1. Response variable and model effect types covered by LMM and GLMM

Replacing the classical model format, the defining elements of the GLMM are as follows:

- Distributions:
  - ○ $y|b \sim \mathcal{D}(\mu, \Sigma)$; $\mathcal{D}$ denotes some distribution, e.g., one of those listed above
  - ○ $b \sim N(0, G)$
  - ○ $f(y|b) \propto exp\left(\frac{y\theta - b(\theta)}{\phi}\right)$, $\mu = E(y|b) = \frac{\partial b(\theta)}{\partial\theta}$, $v(\mu) = \frac{\partial^2 b(\theta)}{\partial\theta^2}$, $Var(y|b) = \phi v(\mu)$

    - $\Sigma = V_{\mu}^{1/2} R V_{\mu}^{1/2}$, $V_{\mu}^{1/2} = diag\left[\sqrt{v(\mu)}\right]$, R is scale matrix, e.g., $R = I\phi$ or residual covariance

- Link function: $\eta = g(\mu)$
- Linear Predictor: $\eta = X\beta + Zb$

PROC MIXED allows you to work in the first row of Table 1. PROCs BGLIMM, GLIMMIX, MCMC and NLMIXED allow you to work with all of the rows.

FREQUENTIST GLMM ESTIMATION AND INFERENCE

The guiding principle of GLMM estimation is maximum likelihood (ML). You obtain estimates of the fixed effects by maximizing the log-likelihood, $log[f(y;\beta)]$, where $f(y;\beta) = \int f(y;\beta|b)f(b)db$. In general, this integral is intractable.

SAS software uses two approximation strategies: linearization (pseudo-likelihood) and integral approximation (quadrature and Laplace). The mixed model equations for Gaussian data, the PROC MIXED default used to obtain REML estimates of the variance components and ML solutions for β and b, are special cases of pseudo-likelihood (PL). The REML version of PL is the PROC GLIMMIX default. Laplace and quadrature are PROC GLIMMIX options. PROC NLMIXED uses quadrature only.

Estimates and standard errors computed from PL or integral approximation are used to compute test statistics, *p*-values and confidence intervals. Classical frequentist inference makes extensive use of significance testing.

BAYESIAN ESTIMATION AND INFERENCE

The primary tool of Bayesian inference is the posterior distribution,

$$f(\beta,\sigma|y,b) = \frac{f(y,b|\beta,\sigma)f(\beta,\sigma)}{\iint f(y,b|\beta,\sigma)f(\beta,\sigma)d\beta d\sigma}$$

where σ denotes the vector of covariance parameters and $f(\beta,\sigma)$ denotes the prior distribution of the fixed effects and covariance parameters. The function $f(y,b|\beta,\sigma)$ is the same likelihood used in frequentist estimation, defined by the GLMM distributions $f(y|b)$ and $f(b)$.

As with the GLMM likelihood, the integrals required for the posterior distribution are generally intractable. Unlike frequentist estimation, neither linearization nor integral approximation are viable options. Instead, Bayesian methods approximate the posterior distribution by simulation.

PROC BGLIMM involves the following steps:

- Specify the GLMM (i.e., the defining elements listed above). This step specifies $f(y|b)$ and $f(b)$.
- Specify the prior distributions $f(\beta,\sigma)$. The BGLIMM procedure uses pre-programmed default priors. Depending on the model and data, you may or may not need to replace them with more appropriate prior distributions.
- PROC BGLIMM implements three computational steps. It is beyond the scope of this tutorial to provide technical description of the algorithms used to implement these steps. The following is simply a list of the steps and their purpose:
    - Tuning. This step uses the data, model and priors to come up with an approximation of the posterior distributions to be sampled.
    - Burn-in. This is initial sampling the posterior distributions. A coin flip exercise in introductory statistics class provides simple analogy: the proportion of flips

resulting in heads vacillates early in the sampling, but eventually settles down to a reasonable approximation of the probability of a head.

  - Sampling the posterior distribution. As the name implies, the posterior distribution of each parameter is sampled with the goal of getting a sufficiently accurate and detailed characterization.

The end result is a mean, standard deviation and quantiles of interest for the posterior distribution of each element of β and each covariance parameter. You can use BGLIMM's ESTIMATE statement to define estimable functions of the β that address objectives of your data analysis. Two forms of credible intervals are commonly used for Bayesian inference:

- the highest posterior density (HPD) interval. This is the narrowest interval between the lower and upper bound that contains a given percent of the posterior distribution.
- quantile-based interval. For example, you can construct a 95% credible interval using the 2.5 and 97.5 percentiles of the posterior distribution as the lower and upper bounds.

There are a number of diagnostics you should check before using PROC BGLIMM results. Diagnostics are introduced in the context of the examples in the following sections. These examples cover problems you are likely to encounter, how to interpret the relevant diagnostics, and PROC BGLIMM options you can use to address these problems in order to obtain a useable analysis.

EXAMPLE 1: MULTI-CLINIC TRIAL WITH BINOMIAL DATA

The first example is the Beitler and Landis (*Biometrics*, 1985) multi-clinic data set that appears in *SAS for Mixed Models* (2018), Chapter 11, Section 11.3. Two treatments, CNTL and DRUG, are compared at eight clinics sampled from a target population. At the j^{th} clinic, n_{ij} patients are assigned to the i^{th} treatment. The response variable, denoted y_{ij}, is the number of patients having a favorable outcome.

The first step in the analysis of these data is to identify an appropriate statistical model. You can do this by following a process presented in *SAS for Mixed Models*, Chapter 5. List the sources of variation separately for the "study" design and the treatment design. The study design, also called the "experiment design," describes the components of the design before being assigned treatments or treatment levels. In this case, the study design consists of clinics and groups of patients within clinics. The treatment design is simply CNTL and DRUG. Groups are randomly assigned to treatments. Table 2 shows the sources of variation.

STUDY DESIGN		**TREATMENT DESIGN**		**COMBINED**	
SOURCE	**DF**	**SOURCE**	**DF**	**SOURCE**	**DF**
clinic	7			clinic	7
		treatment	1	treatment	1
group(clinic)	8			group(clinic) | treatment a.k.a. clinic x treatment a.k.a. unit of observation	8-1=7
TOTAL	**15**			**TOTAL**	**15**

Table 2. Sources of Variation for Multi-Clinic Example Data

In the combined column, read "group(clinic) | treatment" as "group within clinic after accounting for treatment." List the rows so that treatment appears in the row immediately above the unit to which it is assigned – in this case, group(clinic).

The next step is to write model effects associated with each source of variation and the assumed probability distribution of any effects considered to be random. Tables 3 and 4 show two scenarios that lead to plausible mixed models. Table 3 describes the elements of a logit-normal GLMM, so-called because it the uses a logit link function, $log\left[\pi_{ij}/\left(1-\pi_{ij}\right)\right]$, where π_{ij} denotes the probability of a favorable outcome for the ij^{th} clinic-treatment combination, and all random effects are assumed to follow a normal (Gaussian) distribution. Note that clinics effects are random because the clinics in the study are a representative sample. Also note that there is a random effect for the group level source of variation. You should include this term to account for variation in π_{ij} at this level – if you don't, over-dispersion may cause misleading results.

Note that group(clinic) is the unit on which the observations (y_{ij}) are taken. Hence, you can refer to it as the "unit level" effect, unit being shorthand for unit of observation. Specifying the model requires stating the assumed distribution of the observations, conditional on the random effects, at the unit level, and relating its expected value to the linear predictor, e.g., through the inverse link function as shown here.

SOURCE	**EFFECT DISTRIBUTION**	**OBSERVATION AND MODEL**
clinic	$c_j \sim N(0, \sigma_c^2)$	
treatment	τ_i	
group(clinic) | treatment a.k.a. clinic x treatment a.k.a. unit of observation	$ct_{ij} \sim N(0, \sigma_{ct}^2)$	$y_{ij} \vert c_j, ct_{ij} \sim \text{Binomial}\left(N_{ij}, \pi_{ij}\right)$ $\pi_{ij} = 1/\left\{1 + exp\left[-\left(\eta + \tau_i + c_j + ct_{ij}\right)\right]\right\}$

Table 3. Model Effects and Distributions Defining Logit-normal GLMM

Table 4 gives the elements of a beta-binomial mixed model, a commonly used alternative to the logit-normal GLMM for binomial data. The difference between the two models is the way group-level variation in the probability of a favorable outcome is modeled. The logit-normal GLMM includes a random effect in the linear predictor. The beta-binomial models the probability as a random variable, assuming that it follows a beta distribution. Table 4 gives the Ferrari and Cribari-Neto (2004) parameterization of the beta in terms of its mean (μ_{ij}) and scale parameter (φ). Alternatively, you can write the distribution in its conventional math-stat form, $p_{ij} \sim \text{Beta}\left(\alpha_{ij}, \beta_{ij}\right)$, where $\alpha_{ij} = \varphi\mu_{ij}$ and $\beta_{ij} = \varphi\left(1-\mu_{ij}\right)$.

SOURCE	**EFFECT DISTRIBUTION**	**OBSERVATION AND MODEL**
clinic	$c_j \sim N(0, \sigma_c^2)$	
treatment	τ_i	
group(clinic) | treatment a.k.a. clinic x treatment a.k.a. unit of observation	$p_{ij} \sim \text{Beta}\left(\mu_{ij}, \varphi\right)$	$y_{ij} \vert c_j, p_{ij} \sim \text{Binomial}\left(N_{ij}, p_{ij}\right)$ $\mu_{ij} = 1/\left\{1 + exp\left[-\left(\eta + \tau_i + c_j\right)\right]\right\}$

Table 4. Model Effects and Distributions Defining Beta-Binomial Mixed Model

You can use PROC BGLIMM (or PROC GLIMMIX) to implement the logit-normal GLMM, but not the beta-binomial. This is because BGLIMM and GLIMMIX are limited to models with Gaussian random effects, whereas the beta-binomial has a non-Gaussian random effect, p_{ij}. (Although beyond the scope of this tutorial, you can implement the beta-binomial with PROC MCMC)

The beta-binomial is shown here to illustrate the difference between a "sensible" model and an improperly specified model. What constitutes a "sensible" model? You must have a one-to-one correspondence between sources of variation and model effects or parameters that account for them. Table 5 illustrates the difference.

	LOGIT-NORMAL	BETA-BINOMIAL	NAIVE GLMM	BETA-BIN W/ ct_{ij}
SOURCE	sensible	sensible	Over-Dispersion	Unit Confounding
clinic	c_j	c_j	c_j	c_j
treatment	τ_i	τ_i	τ_i	τ_i
unit	ct_{ij}	φ		ct_{ij}, φ

Table 5. Sensible and Improperly Specified Mixed Models for Multi-Clinic Binomial Data

Both the logit-normal and beta-binomial meet the sensible model criterion. The “naive GLMM” lacks any term that accounts for variation at the unit level, making the model vulnerable to over-dispersion. The ct_{ij} effect is thus needed for the logit-normal model, but if you include it in the beta-binomial model, it is confounded with the beta distribution’s scale parameter (φ), which is not good.

The remainder of this section focuses on implementing the logit-normal model with PROC BGLIMM.

BASIC PROC BGLIMM STATEMENTS FOR THE LOGIT-NORMAL BGLIMM

PROC BGLIMM uses syntax borrowed from PROC GLIMMIX and, for repeated measures, from PROC MIXED. To illustrate, here are the basic GLIMMIX and BGLIMM statements. First, the GLIMMIX statements for the logit-normal GLMM from *SAS for Mixed Models*:

```
proc glimmix data=multi_clinic;
 class clinic treatment;
 model fav/nij =  treatment;
 random intercept treatment / subject=clinic;
 lsmeans treatments / ilink diff oddsratio cl;
run;
```

Now the BGLIMM statements:

```
proc bglimm data=multi_clinic plots=(trace autocorr density)
  diagnostics=all outpost=cout;
 class clinic treatment;
 model fav/nij =  treatment / init=pinit;
 random intercept treatment / subject=clinic;
 estimate “CNTL” intercept 1 treatment 1 0;
 estimate “DRUG” intercept 1 treatment 0 1;
 estimate “log_odds_ratio” treatment 1 -1;
 ods output estimates=model_scale;
run;
```

Notice that the CLASS, MODEL and RANDOM statements for the two procedures are identical. The FAV/NIJ (events/trials) syntax is specific to the binomial distribution. Also notice that there are no statements in the basic PROC BGLIMM program specifying prior distributions. BGLIMM has pre-programmed priors that it uses by default. These may or may not be appropriate, depending on the model and data. The additional options in the PROC BGLIMM statement for PLOT and DIAGNOSTICS should be considered standard operating procedure to either verify that the results are useable or to identify problems that need to be fixed, such as inappropriate default priors, before the output can be regarded as useable. The INIT=PINIT option in the MODEL statement causes the default starting values used for the fixed effects to be included is the SAS listing. Consider these to be part of the diagnostics. PROC BGLIMM does not have an LSMEANS statement. You must write the ESTIMATE statements that correspond to least squares means and treatment differences. PROC BGLIMM also lacks an ILINK option. PROC BGLIMM only computes model-scale (in this case, logit scale) estimates. The OUTPOST option in the PROC statement and the ODS OUTPUT statement give you two ways to obtain data-scale statistics. These are explained below in the subsection entitled “Post-Processing to Obtain Data-Scale Statistics.”

DIAGNOSTICS

The PLOT, DIAGNOSTICS=ALL and INIT=PINIT produce items in the SAS listing the help you identify problems and to decide if the results are useable. This section has two parts: a list of the items and a brief description of their purpose, and diagnostics produced by the default PROC BGLIMM statements given above and their interpretation.

List of Diagnostics

The PLOT option gives you three plots: autocorrelation, density and trace. Trace plots are also called "caterpillar" plots, because ideally they should look like a "fuzzy caterpillar," which indicates the sampling algorithm has produced a reasonable approximation of the posterior distribution. The algorithms used to sample the posterior distributions are vulnerable to autocorrelation. The autocorrelation plot shows how quickly autocorrelation decreases as lag increases. The density plot, as the name implies, shows the posterior density produced by the sampling algorithm.

DIAGNOSTICS=ALL produces the following:

- Effective Sample Size (ESS). Autocorrelation reduces posterior sampling efficiency. ESS gives a measure of the sample size after adjusting for autocorrelation. Low efficiency, per se, is not a problem, but ESS should be, at the very least, >1000 in order to have an accurate approximation of the posterior distribution.
- Heidelberger-Welch. These are stationarity tests – is the posterior distribution sampling giving consistent results from beginning (immediately after burn-in) to end? The key columns in the SAS listing will say "passed" or "failed."
- Raftery-Lewis. Addresses the accuracy of the posterior distribution's percentile estimates. The key statistic is the dependence factor. Ideally, it should be ≅ 1.
- Geweke. Compares estimates early and late in the sampling process. Use the Geweke statistics to test the null hypothesis that they are acceptably similar. You want to see *p*-values consistent with failing to reject the null hypothesis.

The listing also includes the number of burn-in iterations, the posterior density sample size and the priors the BGLIMM procedure uses by default.

Results of Preliminary PROC BGLIMM Run with Default Settings

Figure 1 shows two diagnostic plots. On the left is the plot for the logit-normal GLMM's intercept parameter produced by BGLIMM program given above. On the right is an example of what the plot should look like in a run whose results are useable. The plots for the other parameters are not shown, but they are similar to these.

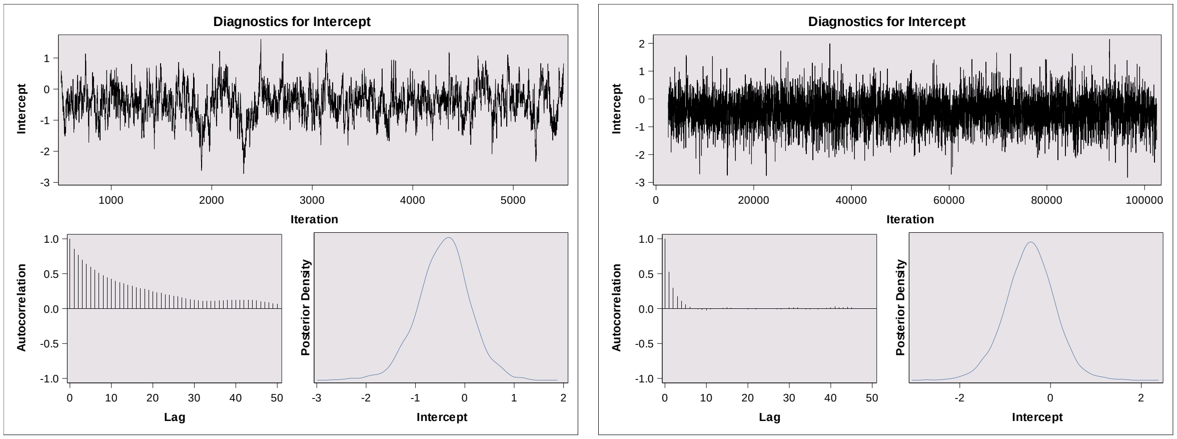


Figure 1. Example Diagnostic Plots

The left-hand plot is what you do not want to see. The trace plot shows erratic and inconsistent sampling of the posterior distribution. It looks more like a seismograph during an earthquake than the "fuzzy caterpillar" plot on the right. The right-hand plot shows consistent sampling from beginning to end. The autocorrelation plot on the left also signals trouble. Autocorrelation should drop to zero quickly, as it does by lag 10 in the right-hand plot. The BGLIMM run with default settings produces autocorrelation that never goes to zero, even at lag 50. If autocorrelation does not drop to zero at least by lag 25, consider the results unusable.

Output 1 shows results produced by the DIAGNOSTICS=ALL option. Statistics that signal trouble are highlighted in **bold**.

Output 1. Diagnostic Statistics from PROC BGLIMM Logit-normal Program with Default Settings

Effective Sample Sizes			
Parameter	ESS	Autocorrelation Time	Efficiency
Intercept	**174.1**	28.7273	**0.0348**
trt cntl	**301.6**	16.5782	**0.0603**
trt drug	**.**	.	**.**
Random VC(1)	**886.7**	5.6389	**0.1773**
Random VC(2)	**722.3**	6.9225	**0.1445**

Geweke Diagnostics		
Parameter	z	Pr > |z|
Intercept	-0.4815	0.6302
trt cntl	-0.2990	0.7649
trt drug	.	.
Random VC(1)	0.5412	0.5884
Random VC(2)	0.9604	0.3369

Raftery-Lewis Diagnostics				
Quantile=0.025 Accuracy=+/-0.005 Probability=0.95 Epsilon=0.001				
	Number of Samples			
Parameter	Burn-In	Total	Minimum	Dependence Factor
Intercept	29	32909	3746	**8.7851**
trt cntl	18	19114	3746	**5.1025**
trt drug	.	.	.	.
Random VC(1)	4	4636	3746	1.2376
Random VC(2)	4	4714	3746	1.2584

Heidelberger-Welch Diagnostics								
	Stationarity Test				Half-Width Test			
Parameter	Cramer-von Mises Stat	p-Value	Test Outcome	Iterations Discarded	Half-Width	Mean	Relative Half-Width	Test Outcome
Intercept	0.0951	0.6092	Passed	0	0.0875	-0.4288	-0.2041	**Failed**
trt cntl	0.2343	0.2098	Passed	0	0.0630	-0.9815	-0.0641	Passed
trt drug	.	.		.	.	.	.	

Heidelberger-Welch Diagnostics								
	Stationarity Test				Half-Width Test			
Parameter	Cramer-von Mises Stat	p-Value	Test Outcome	Iterations Discarded	Half-Width	Mean	Relative Half-Width	Test Outcome
Random VC(1)	0.2008	0.2659	Passed	2000	0.0655	1.8446	0.0355	Passed
Random VC(2)	0.0432	0.9160	Passed	0	0.0855	1.2928	0.0661	Passed

The "Effective Sample Sizes" table ESS and "Efficiency" results indicate trouble. All ESS values are less than 1000. The Geweke statistics look okay here, but you should always check them. The Raftery-Lewis dependence factors for "Random VC(1)" and "VC(2)," the clinic and clinic x treatment variance (σ_c^2 and σ_{ct}^2) respectively, are acceptably close to one, but those for the intercept and "trt cntl" effects (η and τ_{CNTL}) are much too high. Rules of thumb being at best approximate, if you see dependence factors greater than 3 or 4, consider your results to be unusable. Finally, the Heidelberger-Welch half-width test for intercept failed.

ADDRESSING PROBLEMS

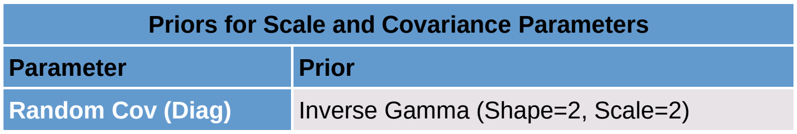
The problems identified by the above diagnostics can occur for several reasons. The number of burn-in iterations or the size of the posterior distribution sampling may be inadequate. You can use thinning, that is, only taking every 5th or 10th or even 50th posterior density sample instead of every sample, to reduce autocorrelation. The default starting values for the fixed effect parameters may not be appropriate. The default priors may be too diffuse, including values that are technically in the parameter space, but highly implausible. Or the default priors may be where the parameters are not, that is, the most likely values of the parameter may be in an extreme tail of the prior distribution. Output 2 shows the defaults used by PROC BGLIMM for the logit-normal GLMM.

Output 2. Sampling, Thinning, Starting Values and Priors for Default Logit-normal BGLIMM Run

Model Information	
Burn-In Size	500
Simulation Size	5000
Thinning	1

Initial Values for Fixed Effects	
Parameter	Value
Intercept	-0.3102
trt cntl	-0.4040

Priors for Fixed Effects	
Parameter	Prior
Intercept	Constant
trt cntl	Constant



Priors for Scale and Covariance Parameters	
Parameter	**Prior**
Random Cov (Diag)	Inverse Gamma (Shape=2, Scale=2)

The “Model Information” table shows the default burn-in size (number of burn-in iterations), the “simulation size” (number of samples of the posterior density) and thinning. Thinning equal one means no thinning was done. The “Initial Values for Fixed Effects” table shows the starting values for η and τ_{CNTL}. Compare these to their estimates from PROC GLIMMIX, $\hat{\eta} = -0.4571$ and $\hat{\tau}_{CNTL} = -0.7462$. You can use the estimates from GLIMMIX as starting values instead of letting BGLIMM’s algorithm choose starting values.

The default priors for η and τ_{CNTL} are “constant” – essentially uniform distributions between plus and minus infinity. Overly diffuse priors that include highly implausible parameter values can cause problems – given that the actual values of η and τ_{CNTL} are unlikely to be very far from their GLIMMIX estimates, the “constant” prior may be an issue. The prior for both variance components is an inverse gamma with shape and scale parameters equal to two. Figure 2 shows a plot of the inverse gamma(2,2) distribution with vertical lines showing the GLIMMIX estimates of σ_c^2 and σ_{ct}^2.

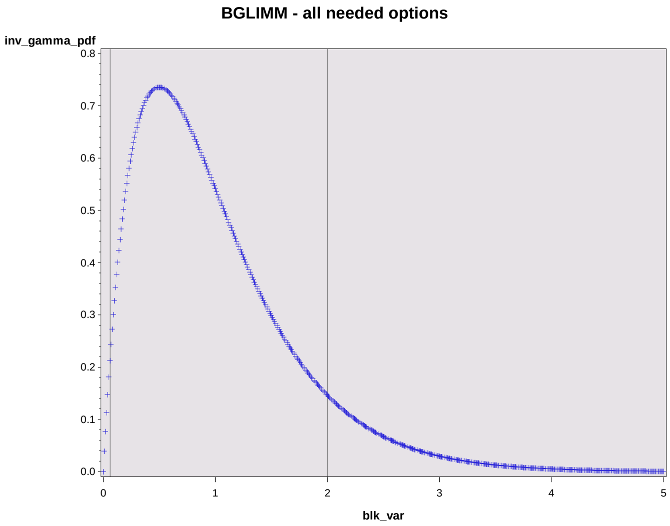


Figure 2. Plot of PROC BGLIMM Default Prior for Variance Components

The lower vertical bar is the GLIMMIX estimate of the clinic x treatment variance, $\hat{\sigma}_{ct}^2 = 0.06$. The upper vertical bar is the GLIMMIX estimate $\hat{\sigma}_c^2 \cong 2$. You can see that the most likely value of the clinic x treatment variance is in the extreme lower tail of the default prior. Most of the probability mass of the inverse gamma(2,2) distribution is located over values that are implausibly high for σ_{ct}^2. The clinic variance shows the opposite problem, although to a lesser extent – its most likely value is in the upper tail of the default prior.

Christiansen, et al. (2011) suggest a strategy for selecting an appropriate prior for a variance component. They suggest using precision, defined as the inverse of the variance, denoted here as $\gamma = 1/\sigma^2$, and identifying a gamma distribution whose mode is approximately equal to the most likely value of the parameter being estimated (you can use the estimates from PROC GLIMMIX as a logical most likely value) and whose upper and lower extreme quantiles (say the 1st and 99th) correspond to values below and above which are thought to be highly unlikely. The mode of a gamma distribution with shape parameter α and scale parameter β is $\beta(\alpha - 1)$. You can specify the mode, try several values of α, solve for β and evaluate the resulting distribution.

The following is an example a program you can use for prior hunting. The example is for the clinic x treatment variance. The PROC GLIMMIX estimate using the pseudo-likelihood default is $\hat{\sigma}^2_{ct} = 0.06$. The estimate using GLIMMIX's METHOD=QUADRATURE option is $\hat{\sigma}^2_{ct} = 0.01$. These translate to precisions of 17 and 100 respectively. The program uses a mode between the two, in this example 40. Use the PDF and CDF functions to create the distribution for each α, β combination. The PROC PRINT statement allows you to examine the lower and upper quantiles of the candidate distributions. The PROC GPLOT statement allows you to visualize the candidate distributions.

The program statements are as follows:

```
data gamma_prior;
 mode=40;
 /* A=shape parameter */
 /* B=scale parameter */
 do a=1.01,1.05,1.1,1.25,1.5,2,3,5,10,20;
  b=mode/(a-1);
  do precision=0 to 200;
   pdf_gamma=pdf("gamma",precision,a,b);
   cdf_gamma=cdf("gamma",precision,a,b);
  output;
 end;
 end;
proc sort data=gamma_prior; by a;
proc print;
 where (0.009<cdf_gamma<0.011 or 0.989<cdf_gamma<0.991);
run;
proc gplot data=gamma_prior;
 by a;
 plot pdf_gamma*precision/href=17,40,100;
 plot cdf_gamma*precision/vref=0.01,0.99;
run;
```

Some trial and error, and art along with science, are involved at this point. The distribution selected is gamma with shape=3 and scale=20. Figure 3 shows plots of the p.d.f. and c.d.f.

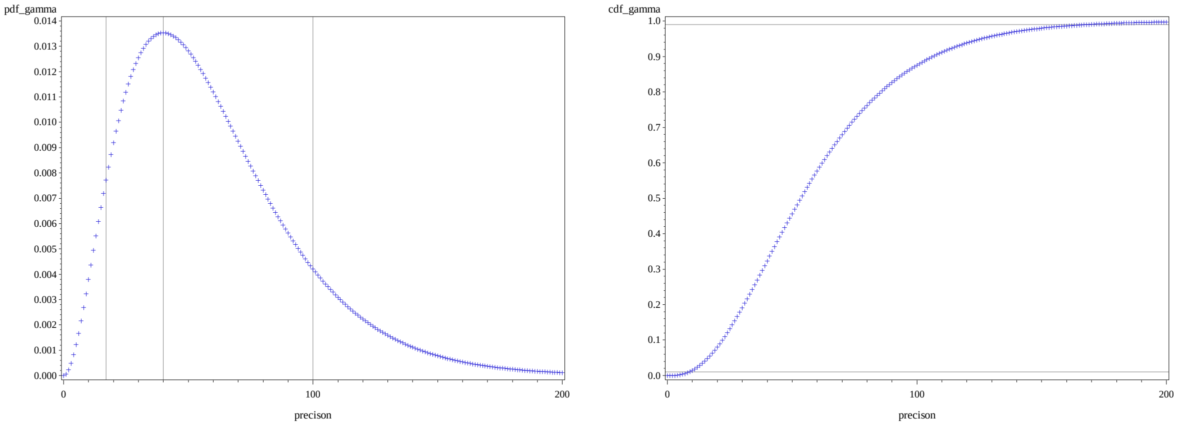


Figure 3. Plots of the Prior Distribution Selected for the Clinic x Treatment Variance

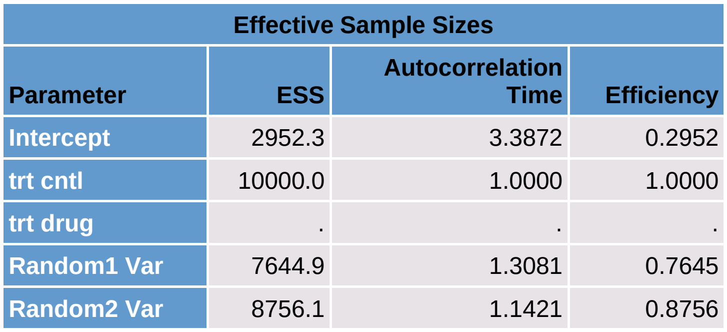
You can see that p.d.f. has a mode of 40 and contains the precision values 17 and 100 well within its range. The c.d.f. plot shows that the lower and upper extreme quantiles are just under 10 and just under 200, respectively. Translated to the variance scale, this prior covers a range between $\sigma^2_{ct} = 0.005$ to σ^2_{ct} just over 0.1, the range considered plausible. Using the same strategy, a suitable prior for the clinic precision is gamma with shape=1.5 and scale=1.

REVISED PROC BGLIMM PROGRAM

The following shows the PROC BGLIMM program modified to include the options needed to obtain useable results. These options appear in **bold**. The program:

```
  proc bglimm data=clinics seed=81152097
     nbi=2500 nmc=100000 thin=10
     plots=(trace autocorr density) diagnostics=all
     statistics(percent=(2.5 50 97.5))
     outpost=cout;
   class clinic trt;
   model fav/Nij=trt /
     init=(list=(-0.46 -0.75) pinit) coeffprior=normal(var=9);
   random  intercept  / subject=clinic covprior=igamma(shape=1.5, scale=1);
   random  trt / subject=clinic covprior=igamma(shape=3, scale=0.05);
   estimate "CNTL" intercept 1 trt 1 0;
   estimate "DRUG" intercept 1 trt 0 1;
   estimate "log odds-ratio" trt 1 -1;
   ods output estimates=modelscale;
   title "BGLIMM - all needed options";
  run;
```

The NBI, NMC and THIN options increase burn-in iterations, posterior distribution sample size and thinning, respectively. The INIT=(LIST=(option allows you to specify starting values for the fixed effect parameters η and τ_{CNTL}. The COEFFPRIOR option specifies a $N(\mu = 0, \sigma^2 = 9)$ prior distribution for these effects. Note that your options for COEFFPRIOR are either “constant” or “normal.” VAR allows you to specify the variance, but the normal prior is always centered at a mean of zero. The COVPRIOR option specifies a prior for the variance of random model effects. Notice that you can have multiple RANDOM statements, which allows you to use different priors for each variance. The COVPRIOR must be specified in terms of the inverse gamma distribution. The required syntax is IGAMMA (SHAPE=$<$ $value$ $>$, SCALE=$<$ $value$ $>$). Use the following result: $\gamma \sim Gamma(\alpha, \beta)$ is equivalent to $1/\gamma \sim inverse - gamma(\alpha, 1/\beta)$. For example, the $gamma(3,20)$ prior for the clinic x treatment precision translates to an $inverse - gamma(3,0.05)$ prior for the clinic x treatment variance. Output 3 shows selected results.



Effective Sample Sizes			
Parameter	ESS	Autocorrelation Time	Efficiency
Intercept	2952.3	3.3872	0.2952
trt cntl	10000.0	1.0000	1.0000
trt drug	.	.	.
Random1 Var	7644.9	1.3081	0.7645
Random2 Var	8756.1	1.1421	0.8756

Raftery-Lewis Diagnostics				
Quantile=0.025 Accuracy=+/-0.005 Probability=0.95 Epsilon=0.001				
	Number of Samples			Dependence Factor
Parameter	Burn-In	Total	Minimum	
Intercept	6	6350	3746	1.6951
trt cntl	2	3834	3746	1.0235
trt drug	.	.	.	.
Random1 Var	2	3803	3746	1.0152
Random2 Var	2	3650	3746	0.9744

Posterior Summaries and Intervals					
Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Intercept	10000	-0.4491	0.5504	-1.6003	0.5728
trt cntl	10000	-0.7454	0.3053	-1.3201	-0.1303
trt drug	0	.	.	.	.
Random1 Var	10000	2.1165	1.5208	0.3744	4.6572
Random2 Var	10000	0.0261	0.0281	0.00444	0.0639

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	2.5	50	97.5
Intercept	10000	-0.4491	0.5504	-1.5574	-0.4418	0.6304
trt cntl	10000	-0.7454	0.3053	-1.3539	-0.7410	-0.1619
trt drug	0	.	.	.	.	.
Random1 Var	10000	2.1165	1.5208	0.6293	1.7386	5.7644
Random2 Var	10000	0.0261	0.0281	0.00701	0.0193	0.0845

Results from ESTIMATE Statements				
Label	Mean	Standard Deviation	95% HPD Interval	
CNTL	-1.1946	0.5612	-2.3136	-0.0881
DRUG	-0.4491	0.5504	-1.6003	0.5728
log odds-ratio	-0.7454	0.3053	-1.3201	-0.1303

Output 3. Selected Results from Useable PROC BGLIMM Run for Beitler-Landis Data

The “Effective Samples Sizes” table shows acceptable ESS and efficiency numbers. All Raftery-Lewis dependence factors are <2. In the interest of space, the other diagnostics are not shown here, but they are similarly acceptable, indicating that this run has produced useable results. The “Posterior Summaries and Intervals” table shows the mean, standard deviation and 95% highest posterior density (HPD) intervals for the model fixed effects and the variance components. The estimates in the “Mean” column are similar to those obtained with PROC GLIMMIX. Note that the fixed effect component of the linear predictor, $\eta + \tau_i$ is not full rank. PROC BGLIMM uses the same convention – setting the last effect, in this case

τ_{DRUG}, to zero – as other SAS linear model procedures. The "Posterior Summaries" table gives percentiles specified by the STATISTICS(PERCENT= option instead of HPD intervals. The "Results from ESTIMATE Statements" table give the posterior distribution mean, standard deviation and 95% HPD intervals from the ESTIMATE statements.

Note that all of these statistics are on the model scale – in this case the logit scale. Because PROC BGLIMM has no ILINK option, you must output these results and use post-processing steps to obtain data scale estimates. The most obvious of these are the probabilities of a favorable outcome for each treatment, $\hat{\pi}_{CNTL}$ and $\hat{\pi}_{DRUG}$, the difference between the two, $\hat{\pi}_{DRUG} - \hat{\pi}_{CNTL}$, and the odds-ratio, $\frac{\hat{\pi}_{CNTL}}{1-\hat{\pi}_{CNTL}} / \frac{\hat{\pi}_{DRUG}}{1-\hat{\pi}_{DRUG}}$. There are two ways to do this. These are shown in the next section.

POST-PROCESSING TO OBTAIN DATA-SCALE ESTIMATE

You can either use the data set produced by the OUTPOST option in the PROC statement or the ESTIMATES from the ODS OUTPUT statement to obtain data-scale estimates.

Data Scale Estimates Using OUTPOST and the %SUMINT Macro

After you run the PROC BGLIMM program that produces useable output, use the following statements:

```
data datasc;
 set cout;
 pr_cntl=logistic(cntl);
 pr_drug=logistic(drug);
 ProbDiff=logistic(drug)-logistic(cntl);
 OddsRatio=exp(log_odds_ratio);
run;
%sumint(data=datasc, var=pr_cntl: pr_drug: ProbDiff: OddsRatio)
```

The OUTPOST option creates a data set of posterior samples for the model fixed effects and ESTIMATES. In this example, the data set is called COUT. The DATA step creates a new data set, DATASC, with the favorable outcome probabilities, labelled PR_CNTL and PR_DRUG, their difference, PROBDIFF, and the odds-ratio. The LOGISTIC function implements the logit's inverse link, $\pi = 1/[1 + exp(-\eta)]$. The difference $\hat{\tau}_{CNTL} - \hat{\tau}_{DRUG}$ estimates the log of the odds-ratio, so $exp(\hat{\tau}_{CNTL} - \hat{\tau}_{DRUG})$ gives you the estimated odds-ratio. The %SUMIT macro computes and prints inferential statistics for the terms listed after VAR= for the data set given by DATA=<*data set name*>. Output 4 shows the results.

Posterior Summaries and Intervals					
Parameter	N	Mean	Standard Deviation	95% HPD Interval	
OddsRatio	10000	0.4970	0.1532	0.2251	0.8004
pr_cntl	10000	0.2461	0.0990	0.0750	0.4475
pr_drug	10000	0.3965	0.1224	0.1679	0.6394
ProbDiff	10000	0.1504	0.0667	0.0248	0.2800

Output 4. Data Scale Estimates from OUTPOST and %SUMINT

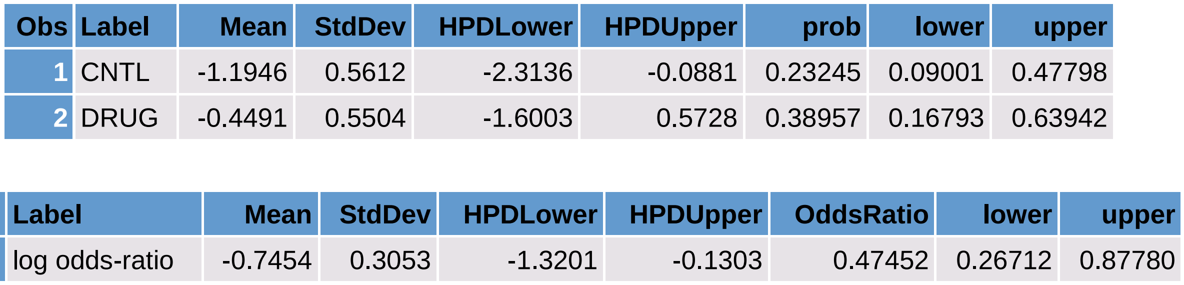
Data Scale Estimates Using ODS OUTPUT

Instead of using the OUTPOST data set and the %SUMINT macro, you can use the following statements:

```
data lsm; set modelscale;
```

```
    if label='CNTL' or label='DRUG';
    prob=1/(1+exp(-Mean));
    lower=1/(1+exp(-HPDLower));
    upper=1/(1+exp(-HPDUpper));
   data oddsratio; set modelscale;
    if label='log odds ratio';
    OddsRatio=exp(Mean);
    lower=exp(HPDLower);
    upper=exp(HPDUpper);
   proc print data=lsm;
   proc print data=oddsratio;
   run;
```

MODELSCALE is the data set produced by the ODS OUTPUT statement. These DATA steps produce two data sets. The one called LSM uses the logit's inverse link to compute the estimates and HPD interval lower and bounds for π_{CNTL} and π_{DRUG}. The data set called ODDSRATIO computes the estimates and HPD bounds for the odds-ratio. Use the IF LABEL= statement to select items to be included in each data set. The LABEL names are the labels used in the ESTIMATE statements. Both data sets use the results of the ESTIMATE statements instead of the posterior sample data set, so the results differ from the %SUMINT listing above. If you want the estimated difference between the probabilities, $\hat{\pi}_{DRUG} - \hat{\pi}_{CNTL}$ use OUTPOST and %SUMINT – it is more convenient. Output 5 shows the results using the ODS OUTPUT approach.



Obs	Label	Mean	StdDev	HPDLower	HPDUpper	prob	lower	upper
1	CNTL	-1.1946	0.5612	-2.3136	-0.0881	0.23245	0.09001	0.47798
2	DRUG	-0.4491	0.5504	-1.6003	0.5728	0.38957	0.16793	0.63942

Obs	Label	Mean	StdDev	HPDLower	HPDUpper	OddsRatio	lower	upper
1	log odds-ratio	-0.7454	0.3053	-1.3201	-0.1303	0.47452	0.26712	0.87780

Output 5. Data Scale Estimates from ODS OUTPUT and Follow-up DATA Steps

The Mean, StdDev, HPDLower and HPDUpper values are model scale values from the BGLIMM ESTIMATE statements. The three columns on the right, labelled "prob" or "OddsRatio" and "lower" and "upper" are the data scale estimates.

EXAMPLE 2: MULTI-LEVEL DESIGN WITH COUNT DATA

This example appears in *SAS for Mixed Models* (2018) in Chapter 13, Section 13.3. The data are from an agricultural experiment comparing two cultivation methods and two seed mixes. Each of six fields is divided into two sections, called whole plots. Each whole plot is randomly assigned to a method so that both methods appear at each field. Each whole plot is divided into two split plots to which mixes are randomly assigned. Table 6 shows the sources of variation.

EXPERIMENT DESIGN		TREATMENT DESIGN		COMBINED	
SOURCE	**DF**	**SOURCE**	**DF**	**SOURCE**	**DF**
field	5			field	5
		method	1	method	1
whole-plot(field)	6			field x method (wp)	6-1=5
		mix	1	mix	1
		method x mix	1	method x mix	1

split-plot(wp)	12			sp (wp) | method, mix	12-2=10
TOTAL	**23**			**TOTAL**	**23**

Table 6. Sources of Variation for Multi-Level Field Trial

The response variable is a discrete count. *SAS for Mixed Models* describes two "sensible" models, the Poisson-normal GLMM and the negative binomial GLMM. This section presents the latter, as it is the model of choice for these data, and the negative binomial illustrates PROC BGLIMM options that would not be used with the Poisson-normal model.

Table 7 uses the combined sources of variation column from Table 6 to show how the negative binomial arises in the context of this experiment.

SOURCE	**EFFECT DISTRIBUTION**	**OBSERVATION AND MODEL**
field	$f_k \sim N\left(0, \sigma_f^2\right)$	
method	α_i	
field x method (wp)	$w_{ik} \sim N(0, \sigma_w^2)$	
mix	β_j	
method x mix	$\alpha\beta_{ij}$	
sp(wp)|method,mix a.k.a. unit of obs.	$u_{ijk} \sim gamma\left({}^{1}/_{\varphi}, \varphi\right)$	$y_{ijk} \vert f_k, w_{ik}, u_{ijk} \sim Poisson\left(\lambda_{ijk} u_{ijk}\right)$ $\lambda_{ijk} = exp\left(\eta + \alpha_i + \beta_j + \alpha\beta_{ij} + f_k + w_{ik}\right)$

Table 7. Model Effects and Distribution Defining Poisson-Gamma Process

The method and mix effects are denoted by α and β, respectively. The negative binomial arises from a Poisson-gamma process. Variation at the unit of observation level is assumed to follow a gamma distribution with $E\left(u_{ijk}\right) = 1$. If you integrate out the unit level term, $y_{ijk} | f_k, w_{ik} \sim NB\left(\lambda_{ijk}, \varphi\right)$, where NB denotes negative binomial. Thus, the following elements define the GLMM:

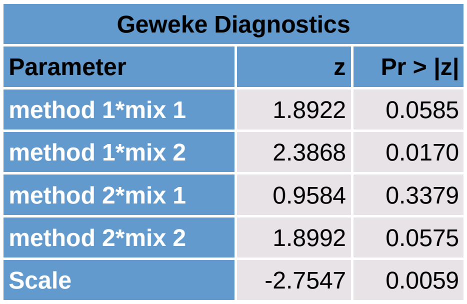
- Distributions: $y_{ijk} | f_k, w_{ik} \sim NB\left(\lambda_{ijk}, \varphi\right)$, $f_k \sim N\left(0, \sigma_f^2\right)$, $w_{ik} \sim N(0, \sigma_w^2)$
- Link function: $\eta_{ijk} = log\left(\lambda_{ijk}\right)$
- Linear predictor: $\eta + \alpha_i + \beta_j + \alpha\beta_{ij} + f_k + w_{ik}$

With PROC BGLIMM, replacing the effects part of the linear predictor, $\eta + \alpha_i + \beta_j + \alpha\beta_{ij}$, with its cell means, full rank equivalent makes it easier to use the ESTIMATE statement to compute terms of interest. The cells means form can be written $\eta_{ij} + f_k + w_{ik}$.

The CLASS, MODEL and RANDOM statements for this model are:

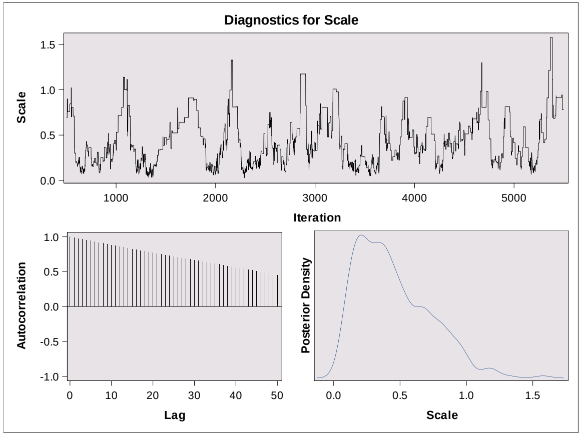
```
  class field method mix;
   model count=method*mix / noint distribution=negbin;
   random intercept method/ subject=field;
```

As with Example 1, the default BGLIMM settings give unusable results. In the interest of space, not all diagnostics are shown. Output 6 shows two examples, the Geweke statistics and the plots for the negative binomial scale parameter, φ.



Geweke Diagnostics		
Parameter	**z**	**Pr > |z|**
method 1*mix 1	1.8922	0.0585
method 1*mix 2	2.3868	0.0170
method 2*mix 1	0.9584	0.3379
method 2*mix 2	1.8992	0.0575
Scale	-2.7547	0.0059

Geweke Diagnostics		
Parameter	z	Pr > |z|
Random VC(1)	1.4784	0.1393
Random VC(2)	-0.4486	0.6537



Output 6. Example diagnostics: PROC BGLIMM negative binomial GLMM with default settings

Four of the seven parameters on the "Geweke Diagnostics" table show low *p*-values. For the scale parameter, *p*<0.01. The trace and autocorrelation plots are unacceptable. To obtain useable results, you must work through the same set of steps as illustrated in the previous section. The following program statements are the result:

```
proc bglimm data=sp_counts plots=(trace autocorr density)
   nbi=2500 nmc=1000000 thin=100 seed=20210209
   diagnostics=all dic outpost=model_scale;
 class field method mix;
 model count=method*mix / noint distribution=negbin
   init=(list=(2.5, 2.9, 0.8, 2.0) Pinit)
   scaleprior=gamma(shape=20,iscale=25);
 random intercept method/ subject=field
   covprior=igamma(shape=4,scale=3);
 estimate "LSM_11" method*mix 1 0 0 0;
 estimate "LSM_12" method*mix 0 1 0 0;
 estimate "LSM_21" method*mix 0 0 1 0;
 estimate "LSM_22" method*mix 0 0 0 1;
 estimate "interaction" method*mix 1 -1 -1 1;
 estimate "Method Main Effect" method*mix 1 1 -1 -1 / divisor=2;
 estimate "Mix Main Effect" method*mix 1 -1 1 -1 / divisor=2;
 ods output estimates=link_scale;
 title "Negative Binomial GLMM";
run;
```

The burn-in, posterior sampling and thinning options may look extreme, but computing time is minimal, and they guarantee useable results. The INIT option uses PROC GLIMMIX results as starting values for the METHOD*MIX (η_{ij}) effects. Changing from the "constant" to the "normal" prior has little effect with this model.

The COVPRIOR in the RANDOM statement is based on using estimates from PROC GLIMMIX, $\hat{\sigma}_f^2 = 1.64$ and $\hat{\sigma}_w^2 = 1.38$, as most likely values. Given that they are similar in value, the same inverse gamma with shape=4 and scale =3 is used for both. The SCALEPRIOR option allows you to specify a prior for the negative binomial scale parameter. The search using the Christiansen, et al. approach described in the previous section resulted in selecting a

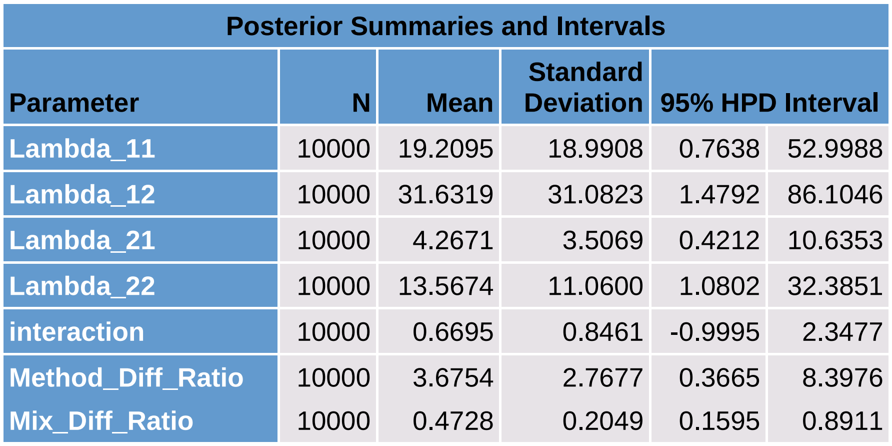
$gamma(shape = 20, scale = 0.04)$ prior. Note that BGLIMM requires the gamma distribution to be specified using the inverse scale, hence ISCALE=25 rather than SCALE=0.04.

The ESTIMATE statements define means, interaction and main effect differences. These are given on the model (log) scale in the SAS listing. You can use either the %SUMINT macro with the OUTPOST=MODEL_SCALE data set or the ODS OUTPUT data set to compute data-scale equivalents. In the interest of space, only the OUTPOST plus %SUMINT program statements and output are shown.

The SAS statements:

```
data data_scale;
 set model_scale;
 Lambda_11=exp(LSM_11);
 Lambda_12=exp(LSM_12);
 Lambda_21=exp(LSM_21);
 Lambda_22=exp(LSM_22);
 Method_Diff_Ratio=exp(Method_Main_Effect);
 Mix_Diff_Ratio=exp(Mix_Main_Effect);
%sumint(data=data_scale, var=Lambda_11: Lambda_12: Lambda_21: Lambda_22:
        interaction: Method_Diff_Ratio: Mix_Diff_Ratio)
```

Output 7 shows the listing.



Posterior Summaries and Intervals					
Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Lambda_11	10000	19.2095	18.9908	0.7638	52.9988
Lambda_12	10000	31.6319	31.0823	1.4792	86.1046
Lambda_21	10000	4.2671	3.5069	0.4212	10.6353
Lambda_22	10000	13.5674	11.0600	1.0802	32.3851
interaction	10000	0.6695	0.8461	-0.9995	2.3477
Method_Diff_Ratio	10000	3.6754	2.7677	0.3665	8.3976
Mix_Diff_Ratio	10000	0.4728	0.2049	0.1595	0.8911

Output 7. Data-Scale Results of Negative Binomial Split-Plot GLMM Analysis

The LAMBDA_11, etc. terms give the rate parameter estimates, $\hat{\lambda}_{ij} = exp(\hat{\eta}_{ij})$ for the four method-mix combinations. The “interaction” term remains on the model scale. There is no need to express it on the data scale, as its only purpose is to assess the magnitude of the method x mix interaction. The HPD interval includes zero; using interval as the criterion, you could conclude that there is insufficient evidence to reject the hypothesis of no interaction. The METHOD_DIFF_RATIO” term estimates $\bar{\lambda}_{1\cdot}/\bar{\lambda}_{2\cdot}$ where $\bar{\lambda}_{i\cdot}$ is the rate parameter for the i^{th} method averaged over both mixes. The difference $\bar{\eta}_{1\cdot} - \bar{\eta}_{2\cdot} = log(\bar{\lambda}_{1\cdot}) - log(\bar{\lambda}_{2\cdot})$, hence $exp(\bar{\eta}_{1\cdot} - \bar{\eta}_{2\cdot}) = \bar{\lambda}_{1\cdot}/\bar{\lambda}_{2\cdot}$. The “MIX_DIFF_RATIO” is similarly defined, but for mix instead of method. Think of these as data scale main effect measures. Although not shown here, you could also obtain statistics for differences between selected LSM terms, similar to the way PROBDIFF was defined in the previous section.

EXAMPLE 3: REPEATED MEASURES DATA

This example is from Chapter 8 of *SAS for Mixed Models* (2018). It appears as “Respiratory Data” for a repeated measures example in Littell, Pendergast and Natarijan (2000), and in Chapter 4 of *SAS for Linear Models, 4th Edition* (Littell, et al., 2002). The data are from a trial comparing a standard asthma treatment (A), a test drug (C) and a placebo (P). Treatments were randomly assigned to 24 patients (72 patients total participated in the trial). The response variable, FEV1, is a measure of breathing ability. A baseline measurement (BASEFEV1) was taken on each patient, then measurements every hour for eight hours starting at one hour after application. FEV1 can be assumed to follow a Gaussian distribution. Table 8 shows the experiment design, treatment design and combined sources of variation.

EXPERIMENT DESIGN		**TREATMENT**		**COMBINED**	
SOURCE	**DF**	**SOURCE**	**DF**	**SOURCE**	**DF**
		drug	2	drug	2
patient	72-1=71			patient(drug) a.k.a. between	71-2=69
		hour	7	hour	7
		drug x hour	14	drug x hour	14
occasion(patient)	72(8-1) = 504			occ(patient)|drug a.k.a. within	504-7-14 = 493
TOTAL	**575**			**TOTAL**	**575**

Table 8. Sources of Variation for Respiratory Repeated Measures Data

In repeated measures terminology, patient(drug) is called the **between subjects** effect and the occ(patient)|drug (occasion within patient after accounting for drug) residual term is called the **within subjects** effect. Following *SAS for Mixed Models*, the LMM of choice has the following elements:

- Distributions
  - $\boldsymbol{y}_{ij}|b_{ij} \sim N(\boldsymbol{\mu}_{ij}, \boldsymbol{R})$, where $\boldsymbol{y}_{ij}' = [y_{ij1} \quad y_{ij2} \quad y_{ij3} \quad \cdots \quad y_{ij7} \quad y_{ij8}]$, the vector of FEV1 measures on the j^{th} patient assigned to the i^{th} drug over the $k = 1,2,3,\ldots,7,8$ hours of observation, $\boldsymbol{\mu}_{ij}$ is the corresponding mean vector and $\boldsymbol{R}$ is the 8×8 covariance matrix modeling serial correlation among the within subjects effects. *SAS for Mixed Models* uses the AR(1) covariance model.
  - $b_{ij} \sim N(0, \sigma_b^2)$ is the between subjects effect
- Linear Predictor: $\mu_{ijk} = \eta + \delta_i + \tau_k + \delta\tau_{ik} + b_{ij}$, where δ and τ denote drug and time (hour) effects, respectively.

The covariance parameters for the for the R matrix are ρ for autocorrelation and σ_w^2 for residual (within subjects) variance. You can include the baseline covariance, BASEFEV1, by modifying the linear predictor to $\mu_{ijk} = \eta + \beta X_{ij} + \delta_i + \tau_k + \delta\tau_{ik} + b_{ij}$, where X_{ij} denotes BASEFEV1 for the j^{th} patient assigned to the i^{th} drug.

Figure 4 shows the interaction plot, which you can obtain using the PLOT=MEANPLOT option in the PROC GLIMMIX LSMEANS statement. The interaction plot is instructive, because it helps you visualize the objectives of this study and the requirements of the model needed to address the objectives.

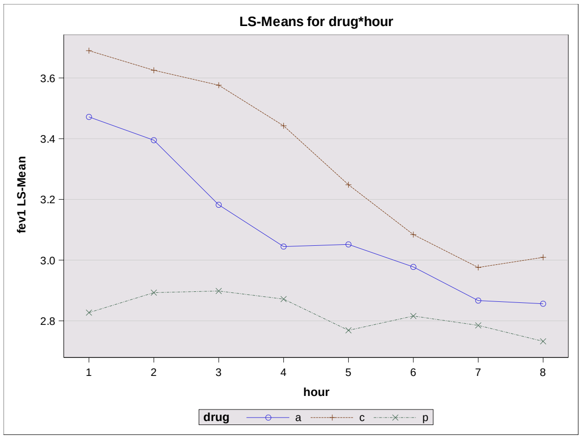


Figure 4. Interaction Plot showing Change in FEV1 over hours by treatment.

The treatment is initially effective for the A and C drugs but not the placebo. Over time, FEV1 decreases as the drugs' effectiveness wears off. The objective of the study is to see if there is a difference between the treatments as measured by initial effectiveness and by how long the effect lasts.

There are two ways to approach this objective. One is to model HOUR as a CLASS variable and track the simple effects $\mu_{Ak} - \mu_{Pk}$ and $\mu_{Ck} - \mu_{Pk}$ at each hour k, where $\mu_{ik} = \eta + \delta_i + \tau_k + \delta\tau_{ik}$. The other is to redefine μ_{ij} in terms of an unequal slopes linear regression model, i.e., $\mu_{ik} = \alpha_i + \beta_i H_k$, where H_k denotes the k^{th} hour, and compare intercept and slope coefficients.

NOTE: The documentation for PROC BGLIMM uses this data set as an example. In one passage, the documentation gives what it calls "Model 2" and "Model 3." The MODEL statements for each are:

- Model 2: `fev1=basefev1 drug hour;`
- Model 3: `fev1=basefev1 drug|hour;`

The documentation characterizes the difference between model 2 and model 3 as "minor," and shows results for model 2 but not model 3. Perhaps the difference is "minor" from a programming point of view, but the difference is not "minor" from a statistical practice point of view. Model 2 is incapable of addressing the objectives of this study and therefore fails the "sensible model" criterion. To address this study's objectives, the drug x hour interaction must be included in the model in some form.

The PROC BGLIMM statements for simple effects and unequal slopes approaches are given as follows. For the simple effects approach, it is much more convenient to specify the cell-means form of the model. The statements are:

```
proc bglimm data=fev1uni nbi=1000 nmc=50000 thin=5 seed=97816352
      plots=(trace autocorr density) diagnostics=all;
   class Drug Patient Hour;
   model FEV1 = BaseFev1 Drug*Hour;
   random int / subject=Patient(drug);
   repeated Hour / subject=Patient(Drug) type=ar(1) r rcorr;
run;
```

Notice that PROC BGLIMM barrows the syntax of the PROC MIXED REPEATED statement to specify the covariance model for the within subjects effect. The default priors work well for this model, but increasing burn-in, posterior sampling and thinning improves the results. There are twenty-four DRUG*HOUR least squares means, and sixteen simple effects comparing drug A and C to the placebo at each hour, for a total to forty ESTIMATE statements. In the interest of space, only one example of each type of statement is shown.

You can access the SAS files for all examples in this tutorial at ... (I need to find out from SAS where...)

The example ESTIMATE statements are:

```
  estimate "LSM A at hour 1"  intercept 1 basefev1 2.6493
     Drug*Hour 1 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0;
  estimate "A vs P at hour 1"
     Drug*Hour 1 0 0 0 0 0 0 0  0 0 0 0 0 0 0 0  -1 0 0 0 0 0 0 0;
```

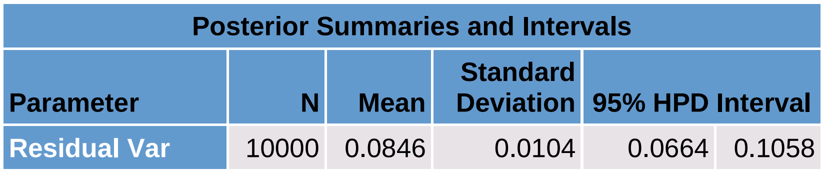
The coefficients for least squares means (LSM) obtain the adjusted mean at the average BASEFEV1 covariate value of 2.6493. If you are in doubt, use the E option of the PROC GLIMMIX LSMEANS statement to get the coefficients needed for these statements.

The statements for the unequal slopes approach are as follows:

```
  data fev1uni;
   set fev1uni;
   H=hour;
  proc bglimm data=fev1uni nbi=1000 nmc=50000 thin=5 seed=44672057
        plots=(trace autocorr density) diagnostics=all;
     class Drug Patient Hour;
     model FEV1 = BaseFev1 drug H(drug);
     random int / subject=Patient(drug);
     repeated Hour / subject=Patient(Drug) type=ar(1) r rcorr;
     estimate "intercept - Drug A" intercept 1 basefev1 2.6493 drug 1 0 0;
     estimate "intercept - Drug C" intercept 1 basefev1 2.6493 drug 0 1 0;
     estimate "intercept - Drug P" intercept 1 basefev1 2.6493 drug 0 0 1;
     estimate "slope - Drug A" H(drug) 1 0 0;
     estimate "slope - Drug C" H(drug) 0 1 0;
     estimate "slope - Drug P" H(drug) 0 0 1;
     estimate "intercept A vs C" drug 1 -1 0;
     estimate "intercept A vs P" drug 1 0 -1;
     estimate "intercept C vs P" drug 0 1 -1;
     estimate "slope A vs C" H(drug) 1 -1 0;
     estimate "slope A vs P" H(drug) 1 0 -1;
     estimate "slope C vs P" H(drug) 0 1 -1;
  run;
```

You need to define a second variable for HOUR – in this case the DATA step with H=HOUR – and use one as the direct regression variable in the MODEL statement and the other as a CLASS variable for the REPEATED statement. The "INTERCEPT – DRUG A" etc. ESTIMATE statements obtain $\hat{\alpha}_i$ the estimated regression intercepts, adjusted for the baseline covariate, and the "SLOPE – DRUG A" etc. statements obtain $\hat{\beta}_i$, the slopes. The next six statements compare intercepts and slopes among the DRUG treatments.

Output 8 shows the covariance parameter estimates obtained using the cell means model. Output 9 shows the least squares means listing from the cell means model. Output 10 shows the simple effect difference results.



Posterior Summaries and Intervals					
Parameter	**N**	**Mean**	**Standard Deviation**	**95% HPD Interval**	
Residual Var	10000	0.0846	0.0104	0.0664	0.1058

Posterior Summaries and Intervals

Parameter	N	Mean	Standard Deviation	95% HPD Interval	
Residual AR(1)	10000	0.5386	0.0550	0.4278	0.6407
Random Var	10000	0.2458	0.0453	0.1612	0.3341

Output 8. Covariance parameter estimates.

"Residual Var" gives the within subjects variance statistics. "Mean" gives you $\hat{\sigma}_w^2 = 0.0846$. "Residual AR(1)" give the autocorrelation parameter, $\hat{\rho} = 0.5386$. "Random Var" gives the between subjects variance, $\hat{\sigma}_b^2 = 0.2458$.

Results from ESTIMATE Statements

Label	Mean	Standard Deviation	95% HPD Interval	
LSM A at hour 1	3.4739	0.1178	3.2501	3.7107
LSM C at hour 1	3.6903	0.1187	3.4705	3.9330
LSM P at hour 1	2.8264	0.1183	2.5956	3.0589
LSM A at hour 2	3.3963	0.1175	3.1685	3.6287
LSM C at hour 2	3.6251	0.1194	3.3849	3.8521
LSM P at hour 2	2.8916	0.1176	2.6606	3.1195
LSM A at hour 3	3.1839	0.1181	2.9470	3.4114
LSM C at hour 3	3.5764	0.1190	3.3390	3.8059
LSM P at hour 3	2.8974	0.1186	2.6571	3.1239
LSM A at hour 4	3.0460	0.1175	2.8098	3.2714
LSM C at hour 4	3.4424	0.1190	3.2178	3.6824
LSM P at hour 4	2.8713	0.1180	2.6237	3.0877
LSM A at hour 5	3.0533	0.1185	2.8278	3.2873
LSM C at hour 5	3.2487	0.1187	3.0172	3.4829
LSM P at hour 5	2.7686	0.1183	2.5390	2.9997
LSM A at hour 6	2.9800	0.1183	2.7533	3.2130
LSM C at hour 6	3.0845	0.1186	2.8482	3.3096
LSM P at hour 6	2.8160	0.1179	2.5817	3.0423
LSM A at hour 7	2.8687	0.1181	2.6414	3.0987
LSM C at hour 7	2.9764	0.1181	2.7273	3.1924
LSM P at hour 7	2.7859	0.1180	2.5595	3.0183
LSM A at hour 8	2.8581	0.1171	2.6377	3.0913
LSM C at hour 8	3.0093	0.1175	2.7724	3.2359
LSM P at hour 8	2.7330	0.1188	2.4999	2.9644

Output 9. Drug x Treatment Least Squares Means for Repeated Measures Data

Results from ESTIMATE Statements				
Label	Mean	Standard Deviation	95% HPD Interval	
A vs P at hour 1	0.6475	0.1652	0.3224	0.9630
C vs P at hour 1	0.8639	0.1683	0.5235	1.1894
A vs P at hour 2	0.5048	0.1648	0.1606	0.8070
C vs P at hour 2	0.7335	0.1678	0.4078	1.0694
A vs P at hour 3	0.2865	0.1651	-0.0478	0.6028
C vs P at hour 3	0.6789	0.1687	0.3453	1.0093
A vs P at hour 4	0.1747	0.1649	-0.1565	0.4920
C vs P at hour 4	0.5711	0.1680	0.2481	0.9090
A vs P at hour 5	0.2847	0.1658	-0.0489	0.6031
C vs P at hour 5	0.4801	0.1677	0.1611	0.8200
A vs P at hour 6	0.1641	0.1651	-0.1476	0.4993
C vs P at hour 6	0.2685	0.1669	-0.0668	0.5785
A vs P at hour 7	0.0828	0.1654	-0.2411	0.4083
C vs P at hour 7	0.1905	0.1673	-0.1365	0.5220
A vs P at hour 8	0.1251	0.1659	-0.1932	0.4553
C vs P at hour 8	0.2763	0.1667	-0.0641	0.5925

Output 10. Simple effect differences between drugs (A, C) and placebo (P)

Output 10 is the PROC BGLIMM analog to the results you would get with the SLICEDIFF=HOUR option in PROC GLIMMIX.

Output 11 gives the ESTIMATE statement results from the unequal slopes regression model.

Results from ESTIMATE Statements				
Label	Mean	Standard Deviation	95% HPD Interval	
intercept - Drug A	3.5169	0.1205	3.2852	3.7577
intercept - Drug C	3.8065	0.1197	3.5730	4.0403
intercept - Drug P	2.8859	0.1199	2.6516	3.1233
slope - Drug A	-0.0887	0.0117	-0.1115	-0.0651
slope - Drug C	-0.1057	0.0117	-0.1285	-0.0825
slope - Drug P	-0.0160	0.0119	-0.0390	0.00708
intercept A vs C	-0.2896	0.1690	-0.6066	0.0506
intercept A vs P	0.6310	0.1706	0.3002	0.9703
intercept C vs P	0.9206	0.1703	0.5897	1.2525
slope A vs C	0.0170	0.0166	-0.0166	0.0486
slope A vs P	-0.0727	0.0168	-0.1057	-0.0399
slope C vs P	-0.0897	0.0167	-0.1236	-0.0578

Output 11. Unequal slopes linear regression repeated measures model estimates and differences

You can see that the estimated intercepts for drug A and C and both greater than for the placebo (3.51 and 3.81 versus 2.89) indicating that initial breathing ability is greater for the two drugs than the placebo. On the other hand, the estimated slopes for drug A and C are -0.09 and -0.11 respectively versus -0.02 for the placebo. The placebo's HPD credible interval for slope includes zero, whereas the intervals for drugs A and C do not. The placebo's regression is essentially flat because its initial effect is negligible.

CONCLUSION

In the SAS system hierarchy, PROC BGLIMM occupies a space between PROC GLIMMIX and PROC MCMC. You can use PROC BGLIMM for any GLMM that PROC GLIMMIX can implement. The advantage of the BGLIMM procedure is that it allows to access to Bayesian estimation and inference using the same CLASS, MODEL and RANDOM syntax as PROC GLIMMIX. The main difference between BGLIMM and GLIMMIX is that because Bayesian analysis depends on complex simulation algorithms, there are more diagnostics that must be monitored and more options that you may need to use in order to get a useable analysis.

PROC MCMC uses syntax borrowed from PROC NLMIXED. This means that it does not have a CLASS statement, making it more tedious to specify models, especially effects models with factorial treatment structures. On the other hand, unlike PROC BGLIMM, you can use PROC MCMC to fit nonlinear models, semiparametric models, and mixed models with non-Gaussian random effects, such as beta-binomial and Poisson-gamma models. PROC MCMC also provides more flexibility in specifying prior distributions. For example, you can use priors centered at the starting value for fixed effects, where BGLIMM limits you to "constant" or zero-centered Gaussian priors.

Finally, when you use PROC BGLIMM with non-Gaussian response variables, these is an additional post-processing step, either using the %SUMINT macro or the ODS OUTPUT data set, in order to get data scale estimates inferential statistics. The lack of an ILINK option means an extra step, but as we see in Examples 1 and 2, it is not a difficult step.

PROC BGLIMM makes Bayesian analysis highly accessible to data analysts who have PROC GLIMMIX experience but are new to the Bayesian world.

REFERENCES

Beitler, P.J. and J.R. Landis. 1985. "A Mixed-Effects Model for Categorical Data." *Biometrics*, 41:991-1000.

Christiansen, R., W. Johnson, A. Branscum and T.E. Hanson. 2011. *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. Boca Raton, FL: CRC Press.

Ferrari, S. L. P., and Cribari-Neto, F. 2004. "Beta Regression for Modelling Rates and Proportions." *Journal of Applied Statistics* 31:799–815

Littell, R.C, J. Pendergast and R. Natarajan. 2000. "Modeling Covariance Structure in the Analysis of Repeated Measures Data." *Statistics in Medicine*, 19:1793-1819.

Littell, R.C., W.W. Stroup and R.J. Freund. 2002. *SAS® for Linear Models, 4th edition*. Cary, NC: SAS Institute, Inc.

Stroup, W.W. 2013. *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*. Boca Raton, FL: CRC Press.

Stroup, W.W., G.A. Milliken, E.A. Claassen and R.D. Wolfinger. 2018. *SAS® for Mixed Models: Introduction and Basic Applications*. Cary, NC: SAS Institute Inc.

ACKNOWLEDGMENTS

Acknowledge Oliver Schabenberger, who developed PROC GLIMMIX and was a valued partner in GLMM’s early days. Fang Chen, who developed PROC BGLIMM and who provided helpful advice about the procedure. George Milliken, Russ Wolfinger and Elizabeth Claassen, my *SAS® for Mixed Models* co-authors and esteemed mixed model brain trust.

RECOMMENDED READING

- *SAS® for Mixed Models: Introduction and Basic Applications*
- *Generalized Linear Mixed Models: Modern Concepts, Methods and Applications*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Walt Stroup
wstroup@unl.edu

The next two paragraphs are **required** and must be in the paper if you are a SAS employee who is authoring a SAS paper. Otherwise, this section is not needed.

Paper 2776-2018

Bayesian Networks for Causal Analysis

Fei Wang and John Amrhein, McDougall Scientific Ltd.

ABSTRACT

Bayesian Networks (BN) are a type of graphical model that represent relationships between random variables. The networks can be very complex with many layers of interactions. Graphical models become BNs when the relationships are probabilistic and uni-directional. Building BNs for causal analyses is a natural and reliable way of expressing (and confirming or refuting) our belief and knowledge about cause and effects. In addition, BNs can be easily reconfigured with minor modifications to facilitate our understanding of probabilistic mechanisms. This paper describes the construction of BNs for causal analyses and how to infer causal structures from observational and interventional data. The paper includes applications of causal BNs for classification using the HPBN Classifier node in Enterprise Miner. Visualization, inferences, and scenario analyses for the examples are discussed.

BAYESIAN CONCEPT OVERVIEW

BAYES' THEOREM

Bayes' theorem (Bayes' law or Bayes' rule) describes the probability of an event based on prior knowledge of conditions that might be related to the event. It can be stated mathematically as the following equation:

$$P(A|B) = \frac{P(A,B)}{P(B)} \qquad (1)$$

where A and B are observed events, P(B) is the probability of observing B independently, P(A,B) is a joint probability of A and B which represents the likelihood of event A and B occurring together, and P(A|B) is a conditional probability which represents the likelihood of event A occurring given that B is observed.

An algebraic inversion of (1) can be written as:

$$P(A,B) = P(A|B)P(B) = P(B|A)P(A) \qquad (1')$$

There are other equivalent statements of Bayes' theorem, each being useful in a different context.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \text{or} \quad P(A|B) = P(A)\frac{P(B|A)}{P(B)} \qquad (2)$$

$$P(A|B) = \frac{P(B|A)P(A)}{\sum_i P(B|A_i)P(A_i)} \quad \text{or} \quad P(A|B) = P(A)\frac{P(B|A)}{\sum_i P(B|A_i)P(A_i)} \qquad (3)$$

Equation (1) gives rise to (2). Equation (3) stems from (2) by substituting the unconditional (marginal) probability of event *B* with the sum of its partitions over all possible outcomes of event *A*.

INTERPRETATION

The interpretation of Bayes' theorem depends on the interpretation of probability. In Bayesian interpretation, probability measures a degree of belief; while in frequentist interpretation, probability measures a mathematical frequency of outcomes.

Bayesians interpret Bayes' theorem as: the updated (posterior) belief of event A given that event B has been observed, P(A|B), is dependent upon the prior belief of event A, P(A), the likelihood of event B given that event A has occurred, P(B|A), and the prior belief of event B, P(B).

Frequentists, on the other hand, interpret Bayes' theorem as: over repeated sampling, P(A|B) is the proportion of outcomes with A out of all outcomes with B.

ADVANTAGE VS. DISADVANTAGE

There are a few advantages of using Bayesian concepts and analytical methods: 1) Bayesian methods can incorporate prior information, knowledge, and subject matter expertise without corresponding data; 2) Bayesian methods enable the study of a cause-effect interpretations rather than just correlations.

However, the sometimes-subjective nature of priors is a disadvantage of Bayesian methods. The posterior distribution may be heavily influenced by a subjective prior in some situations. One can use non-informative priors to mitigate the impact of subjectivity on the posterior estimates.

BAYESIAN NETWORK

DEFINITIONS AND PROPERTIES

A Bayesian Network (BN) is a representation of a joint probability distribution of a set of random variables with probabilistic dependencies. It is a class of graphic models that consist of two parts, <**G**, **P**>:

- **G** is a directed acyclic graph (DAG) made up of nodes corresponding to random variables, ***X*** in ***U***, and arcs (edges, links, or connectors) representing conditional dependencies between random variables
- **P** is a set of conditional probability distributions, one for each node conditional on its parents.

Figure 1 shows an example of a DAG. Letters A – G represent nodes. The graph is directed because each arc between two nodes is uniquely directed, and is acyclic because no cycles or loops (e.g. A→B→C→A) exist. A node from which a directed edge starts is called the parent of the node to which the edge is directed; a node on which a directed edge ends is called the child of the node from which it comes. For example, in Figure 1, nodes B and C are parents of node A, and node A is the common child of nodes B and C. In the example for node A, nodes B, C, and F are ancestors of node A; nodes D, E, and H are descendants of node A. Parents and children of a node are the neighbors of the node. Spouses are nodes sharing the same child, such as A and G. Parents, children, and spouses of a node constitute the Markov blanket of the node.

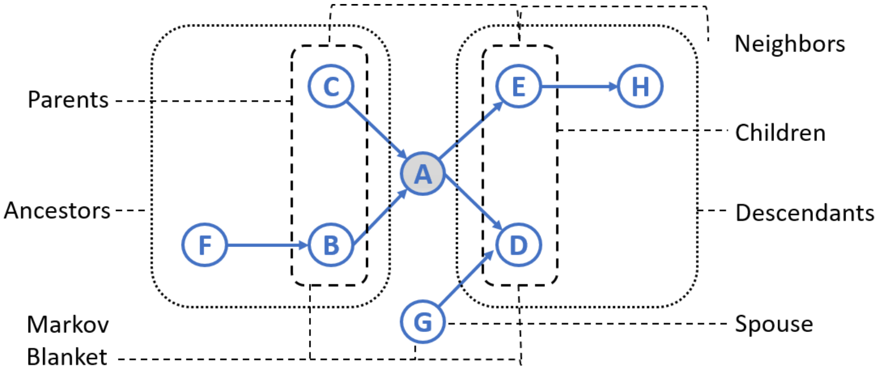


Figure 1 Parents, Ancestors, Neighbors, Spouse, Children, Descendants, and Markov Blanket of Node A in a DAG

The component **P** of a BN is a set of conditional probability distributions, one for each node conditional on its parents. It follows the Markov property of BN: *each node is conditionally independent of its ancestors given its parents*. One can claim that changes in the belief of a node have no impact on the belief of its ancestors given its parents.

The Markov property of BN enables the presentation of the joint probability distribution of random variables in **G** decomposed into a product of conditional probability distributions. For Figure 1, we have

$$P(\mathrm{A,B,C,D,E,F,G,H}) = P(\mathrm{A|B,C})P(\mathrm{B|F})\mathrm{P(C)}P(\mathrm{D|A,G})\mathrm{P(E|A)P(F)P(G)P(H|E)}.$$

It is easy to see that, given the Markov blanket, a node is independent from the rest of nodes. Therefore, the probability distribution of a node is completely determined by its Markov blanket.

Once a BN is learnt from data, one can investigate the effects of a new piece of evidence ***E*** (***E*** ⊆ ***U***) in the distribution of ***U*** using the knowledge encoded in the BN, that is, to investigate the posterior probability P(***Q***|***E***, **G**, **P**), where ***Q*** ∪ ***E*** = ***U***. The posterior probability can be interpreted in terms of the changes in one's beliefs according to the observed new evidence.

BN LEARNING IN SAS®

In many practical settings, the BN is unknown and needs to be learnt from data. The task of fitting a BN is known as BN learning. Given training data and prior information, such as expert knowledge about possible casual relationships, BN learning performed in two steps; structure learning and parameter learning.

Structure Learning

The first step consists in identifying the graph structure of the BN; i.e. how the variables relate to each other as parents, children, neighbors, and spouses. Ideally, the final learnt structure should result in a joint posterior distribution as close as possible to the correct one in the probability space.

The HPBNET procedure deployed in the HPBN node in SAS® Enterprise Miner™ (EM) is a high-performance procedure which can learn different types of BN structures from an input data set. The supported types of network structures are naïve Bayesian (NB), tree-augmented naïve (TAN), Bayesian network-augmented naïve (BAN), parent-child Bayesian network (PC), and Markov blanket (MB). Refer to the reference by Liu et al (2017) for descriptions of each type. Briefly, NB, TAN, and BAN all have one target variable that is a parent to each input variable. NB is similar to a typical linear regression in that the only dependency is between a target variable and the inputs. The inputs themselves are conditionally independent of each other given the target. A TAN extends the NB by allowing a tree structure among the inputs, and a BAN loosens the structure among the inputs to be a BN itself. A PC BN allows inputs to be parents of the target, and the MB is the least restrictive as shown in Figure 1.

Based on the specified BN type, PROC HPBNET uses different algorithms to learn the relationships between the variables, i.e. to define the structure of the graph.

Two categories of algorithms are used in PROC HPBNET:

- The score-based approach uses the Bayesian information criterion (BIC) to measure the goodness of fit of a structure based on the training data, and to find a structure with the highest BIC score. The BIC is a penalized likelihood score. For discrete data, it is defined as

$$BIC = N\sum_{i=1}^{n}\sum_{j=1}^{q_i}\sum_{k=1}^{r_i} p(\pi_{ij})p(X_i = x_{ik}|\pi_{ij})lnp(X_i = x_{ik}|\pi_{ij}) - \frac{M}{2}lnN$$

  where N is the number of observations in the training data, n is the number of variables (nodes), X_i is a random variable, r_i is the number of levels for X_i, x_{ik} is the kth value of X_i, q_i is the number of value combinations of X_i's parents, π_{ij} is the jth value of X_i's parents, and $M = \sum_{i=1}^{n}(r_i - 1)\times q_i$ is the number of parameters for the probability distributions.

- The constraint-based approach uses independence tests to determine the edges and the directions. In some cases when the direction of the edge cannot be oriented, PROC HPBNET uses the BIC score to determine directions of the edges.

When there are many input variables, the number of variable combinations is exponential and structure learning is time consuming. Therefore, variable selection is needed.

Because the probability distribution of a node is completely determined by its Markov blanket, the Markov blanket can be used for variable selection. PROC HPBNET supports two types of variable selection: independence tests and conditional independence tests. Independence tests, such as a chi-square test for discrete data, are performed between each input variables and the target variable. A significant result of the independence test indicates an edge between the input variable and the target variable. While conditional independence tests are performed between each input variable and the target variable given any subset of other input variables.

Take the MB BN structure type for example. PROC HPBNET learns the parents of the target variables first by applying independence tests to determine the edges. The edges are then oriented by using conditional tests and the BIC score. Then PROC HPBNET learns the parents of the input variable which has the highest BIC score with the target by using the same method, and continuous to learn the parents of the input variable which has the second highest BIC score with the target.

Parameter Learning

Parameter learning determines the probability distribution of each node in a BN. Once the structure of the network has been learnt from the data, estimating the parameters of the global distribution is greatly simplified by the application of the Markov property.

One can also classify the value of a target variable using the posterior probability distribution and an observation of other variables. The predicted class is the one that has the largest posterior probability. That is, for an observation of the input variables, $(X_1 = x_1, X_2 = x_2, \cdots, X_{n-1} = x_{n-1})$, the predicted class of target ***T*** is c satisfying

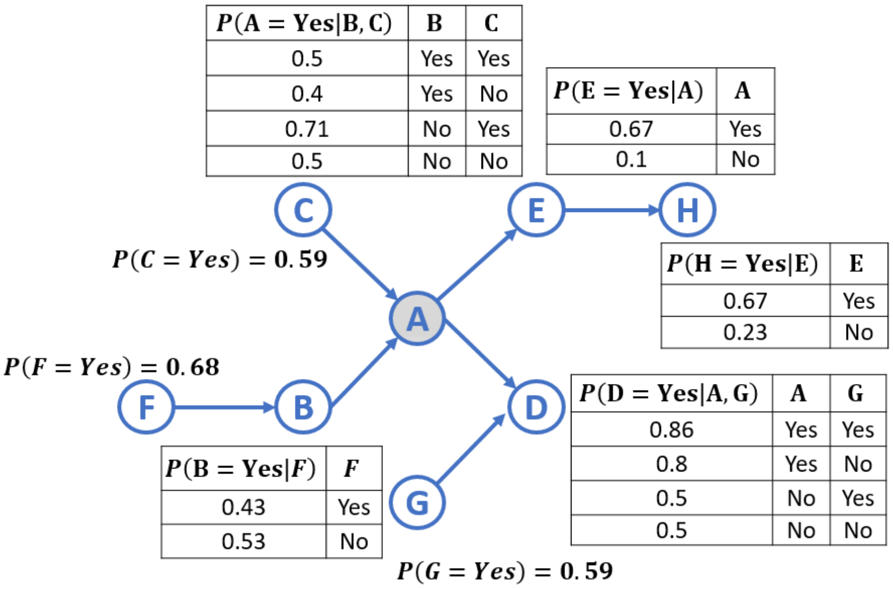
$$max_c P(T = c|X_1 = x_1, X_2 = x_2, \cdots, X_{n-1} = x_{n-1})$$

where

$$\begin{aligned} P(T = c|X_1 = x_1, X_2 = x_2, \cdots, X_{n-1} = x_{n-1}) \quad &\propto P(T = c, X_1 = x_1, X_2 = x_2, \cdots, X_{n-1} = x_{n-1}) \\ &= \prod_{i=1}^{n-1} P\big(X_i = x_i \big| parents(X_i)\big)\, P\big(T = c \big| parents(T)\big)\,. \end{aligned}$$

For example, Figure 2 shows a learnt BN from a dataset in which all variables have a value of either “Yes” or “No”. Suppose we observe an event (B=Yes, C=Yes, D=Yes, E=No, F=Yes, G=No, H=Yes), we want to predict the value of A. Based on the learnt BN in Figure 2, we have

$$\begin{aligned} &P(A = Y|\mathrm{B} = \mathrm{Y}, \mathrm{C} = \mathrm{Y}, \mathrm{D} = \mathrm{Y}, \mathrm{E} = \mathrm{N}, \mathrm{F} = \mathrm{Y}, \mathrm{G} = \mathrm{N}, \mathrm{H} = \mathrm{Y}) \\ &\qquad \propto P(A = Y, \mathrm{B} = \mathrm{Y}, \mathrm{C} = \mathrm{Y}, \mathrm{D} = \mathrm{Y}, \mathrm{E} = \mathrm{N}, \mathrm{F} = \mathrm{Y}, \mathrm{G} = \mathrm{N}, \mathrm{H} = \mathrm{Y}) \end{aligned}$$



$P(\mathbf{A} = \mathbf{Yes}|\mathbf{B}, \mathbf{C})$	B	C
0.5	Yes	Yes
0.4	Yes	No
0.71	No	Yes
0.5	No	No

$P(\mathbf{E} = \mathbf{Yes}|\mathbf{A})$	A
0.67	Yes
0.1	No

$P(\mathbf{H} = \mathbf{Yes}|\mathbf{E})$	E
0.67	Yes
0.23	No

$P(\mathbf{D} = \mathbf{Yes}|\mathbf{A}, \mathbf{G})$	A	G
0.86	Yes	Yes
0.8	Yes	No
0.5	No	Yes
0.5	No	No

$P(\mathbf{B} = \mathbf{Yes}|F)$	F
0.43	Yes
0.53	No

Figure 2 A Bayesian Network

$$= P(A = Y|B = Y, C = Y)P(\mathrm{B = Y|F = Y})\mathrm{P(C = Y)P(D = Y|A = Y, G = N)P(E = N|A = Y)P(F = Y)P(G = N)P(H = Y|E = N)}$$

$$= 0.5\times 0.43\times 0.59\times 0.8\times(1 - 0.67)\times 0.68\times(1 - 0.59)\times 0.23$$

$$= 0.002;$$

And

$$P(A = N|\mathrm{B = Y, C = Y, D = Y, E = N, F = Y, G = N, H = Y})$$

$$\propto P(A = N, \mathrm{B = Y, C = Y, D = Y, E = N, F = Y, G = N, H = Y})$$

$$= P(A = N|B = Y, C = Y)P(\mathrm{B = Y|F = Y})\mathrm{P(C = Y)P(D = Y|A = N, G = N)P(E = N|A = N)P(F = Y)P(G = N)P(H = Y|E = N)}$$

$$= (1 - 0.5)\times 0.43\times 0.59\times 0.5\times(1 - 0.1)\times 0.68\times(1 - 0.59)\times 0.23$$

$$= 0.004.$$

Thus, if an event with (B=Yes, C=Yes, D=Yes, E=No, F=Yes, G=No, H=Yes) is observed, A will most likely happen with a value of No

Note that, the normal constant $P(\mathrm{B = Y, C = Y, D = Y, E = N, F = Y, G = N, H = Y})$ is eliminated from the calculation of the two conditional probabilities of A given other nodes. The exact conditional probability can be obtained by dividing the normal constant which can be easily calculated from the learnt BN.

CAUSAL INFERENCE

Suppose we have learnt a BN with structure **G** and conditional probability distribution **P.**

When a BN is given a causal interpretation, the arcs describe cause-and-effect relationships instead of probabilistic dependencies. For instance, according to Pearl's *Causality*, in a causal BN, the parents of each node are its direct causes, ancestors other than parents are its indirect causes, children of each node are its direct effects, and descendants other than children are its indirect effects.

To evaluate the conditional probability of known causes given their effects and vice versa, three additional assumptions are needed:

- Each node is conditionally independent of its indirect causes, given its direct causes. This assumption is call the causal Markov property, and is the causal interpretation of the Markov property.
- There must exist a network structure which represents the true dependence structure of **G**. This assumption is usually referred as the faithfulness assumption.
- There must be no latent variables which are unobserved variables influencing the variables in the network, acting as confounding factors.

The third assumption is a deduction of the first two. The presence of latent variables may induce wrong correlations between observed variables. If an arc is wrongly added, the causal Markov property might be violated. And, the presence of latent variables violates the assumption of faithfulness.

In this setting, the posterior probability is interpreted as measures of the effects of interventions (denoted as ***I***, and ***I*** ⊆ ***U***) on the causal structure. The intervention represents an action whose only effect is to fix the values of the variables in ***I*** to specific values, such as ***I*** = $(X_{i1} = x_{i1}, X_{i2} = x_{i2}, \cdots, X_{ik} = x_{ik})$, where $i1, \cdots, ik \in \{1, 2, \cdots, n\}$. Then the conditional probability P(***Q***|***I***, **G**, **P**) studies the effect of interventions ***I***.

According to the causal Markov property of the BN, P(***Q***|***I***, **G**, **P**) can be calculated as follows:

$$\mathrm{P}(\boldsymbol{Q}|\boldsymbol{I}, \mathbf{G}, \mathbf{P}) = \mathrm{P}(\mathbf{I})\times\prod P(\boldsymbol{Q}|Parents(\boldsymbol{G}, \boldsymbol{P})),$$

where $Parents(\boldsymbol{G}, \boldsymbol{P})$ is the set of parents for variables in **Q**.

In discrete case of data, $\prod P(\boldsymbol{Q}|Parents(\boldsymbol{G}, \boldsymbol{P}))$ can be obtained by using the relative frequency of **Q** conditional on $Parents(\boldsymbol{G}, \boldsymbol{P})$.

EXAMPLE

The example is a small synthetic data set from Lauritzen and Spiegelhalter (1988) about lung diseases (tuberculosis, lung cancer, and bronchitis), in which the example is motivated as follows:

“Shortness-of-breath (dyspnea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-ray do not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnea.”

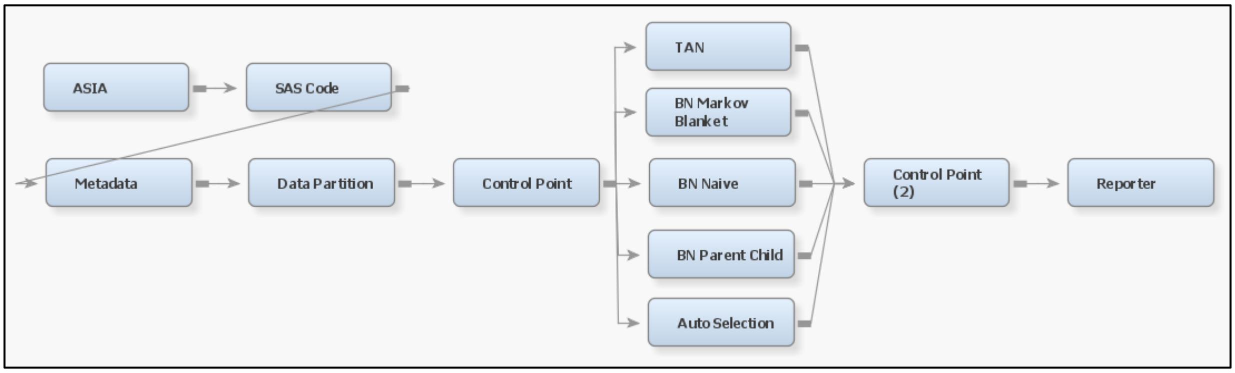
The data set contains 5000 observations and 8 binary character variables (Table 1) whose values are either yes or no. Suppose that we want to determine the dependencies between a set of inputs, dyspnea, bronchitis, a visit to Asia, smoking, and a positive x-ray, and the verified presence of either tuberculosis or lung cancer.

Table 1. Details of the Asia Data Set

Variable	Label	Description	Frequency “Yes”
Dyspnea	Dyspnea	Presence of dyspnea?	2350
Tuber	Tuberculosis	Presence of tuberculosis?	44
Cancer	Lung Cancer	Presence of lung cancer?	330
Bronch	Bronchitis	Presence of bronchitis?	2549
Visit	Visit to Asia	Visit to Asia?	42
Smoke	Smoking	Is a smoker?	2515
XrayPos	Chest X-ray	Positive chest X-ray?	569
TubOrCan	Tuberculosis or Lung Cancer	Presence of tuberculosis or lung cancer?	370

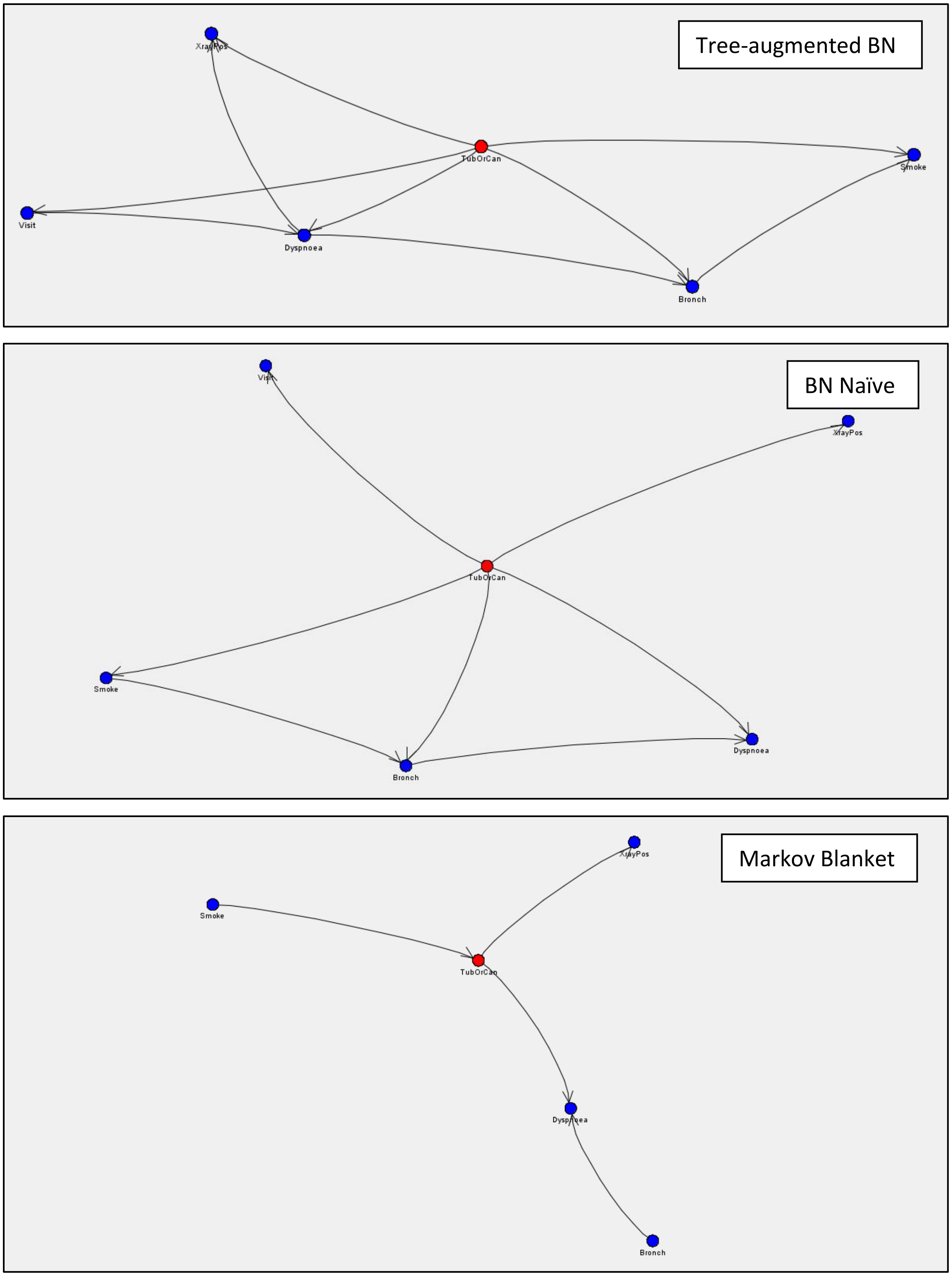
Display 1 shows the EM process diagram. The SAS Code node is used to rename variables and assign labels. The Metadata node sets *TubOrCan* as the target, and the Data Partition node splits the data 3:1 into training and validation data respectively. We configured four structure types, TAN, MB, Naïve, and PC. The fifth HPBN node, Auto Selection, is configured for EM to auto-select the best fitting structure based on criterion such as misclassification rate.

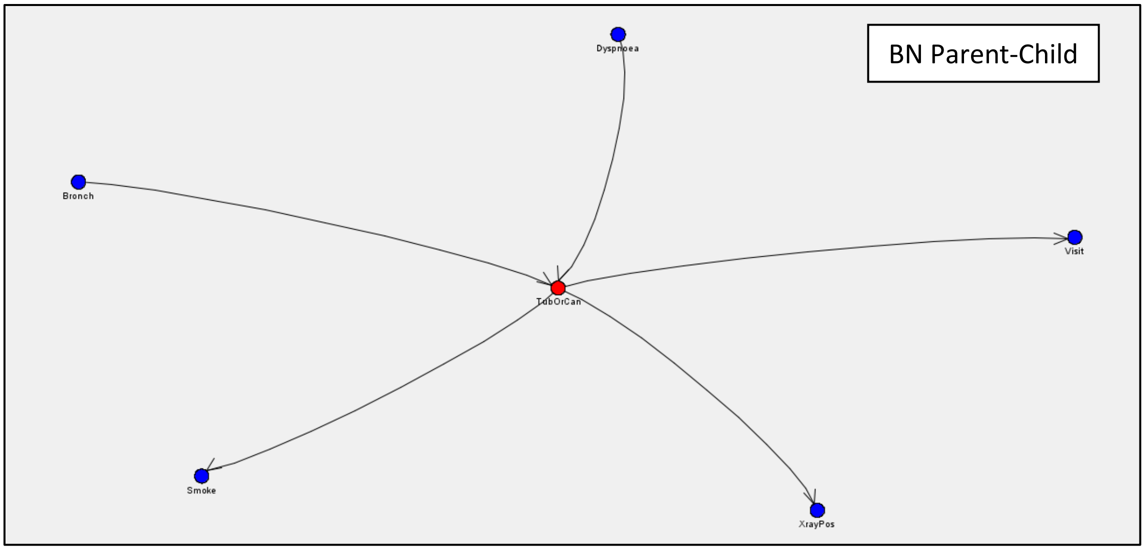
Display 1. Bayesian Network Diagram Based on Asia Data Set



We configured the HPBN nodes to not use the tuberculosis and cancer variables. We also turned off variable pre-screening and selection because we wanted to explore the dependencies among the other inputs. Therefore, our possible models are reduced to the various BN structure types and the edges connecting the nodes. Display 2 shows the final BN structure for each fitted structure type.

Display 2. Four BN Structures for the Asia Data

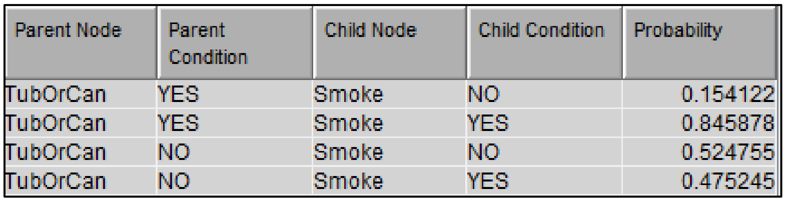




The learnt structures differ because of the restrictions placed on the connections within each type. The HPBN node, which is configured for auto-selection, chose the BN Parent-Child structure. The target variable is always at the center and is shown as red. The inputs are blue and the direction of the arcs are from parent to child. Posterior probabilities are given as P(child | parent). You must consider the temporal relationship between the parent and child, which HPBN will not know, to properly state probable cause for an effect.

For example, the S*moke* node is a child of *TubOrCan*. The posterior conditional probability of smoking (cause) given tuberculosis or cancer (effect) is 0.846 (Output 1). Because we know that smoking pre-dates disease onset, **if this network completely represented all factors influencing disease onset**, then we could conclude, with very high probability, that smoking causes tuberculosis or cancer.

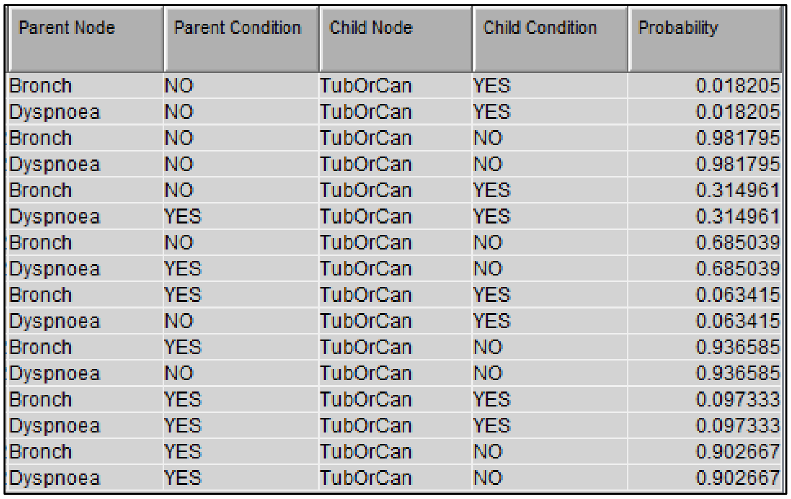
Output 1. BN-PC Posterior Probabilities of Smoking Given Tuberculosis or Cancer



Parent Node	Parent Condition	Child Node	Child Condition	Probability
TubOrCan	YES	Smoke	NO	0.154122
TubOrCan	YES	Smoke	YES	0.845878
TubOrCan	NO	Smoke	NO	0.524755
TubOrCan	NO	Smoke	YES	0.475245

The two parents of *TubOrCan*, bronchitis and dyspnea, may be symptoms of tuberculosis or cancer. Therefore, the posterior conditional probabilities could be used as diagnostic aids.

Output 2. BN-PC Posterior Probabilities of Tuberculosis or Cancer Given Dyspnea and Bronchitis



Parent Node	Parent Condition	Child Node	Child Condition	Probability
Bronch	NO	TubOrCan	YES	0.018205
Dyspnoea	NO	TubOrCan	YES	0.018205
Bronch	NO	TubOrCan	NO	0.981795
Dyspnoea	NO	TubOrCan	NO	0.981795
Bronch	NO	TubOrCan	YES	0.314961
Dyspnoea	YES	TubOrCan	YES	0.314961
Bronch	NO	TubOrCan	NO	0.685039
Dyspnoea	YES	TubOrCan	NO	0.685039
Bronch	YES	TubOrCan	YES	0.063415
Dyspnoea	NO	TubOrCan	YES	0.063415
Bronch	YES	TubOrCan	NO	0.936585
Dyspnoea	NO	TubOrCan	NO	0.936585
Bronch	YES	TubOrCan	YES	0.097333
Dyspnoea	YES	TubOrCan	YES	0.097333
Bronch	YES	TubOrCan	NO	0.902667
Dyspnoea	YES	TubOrCan	NO	0.902667

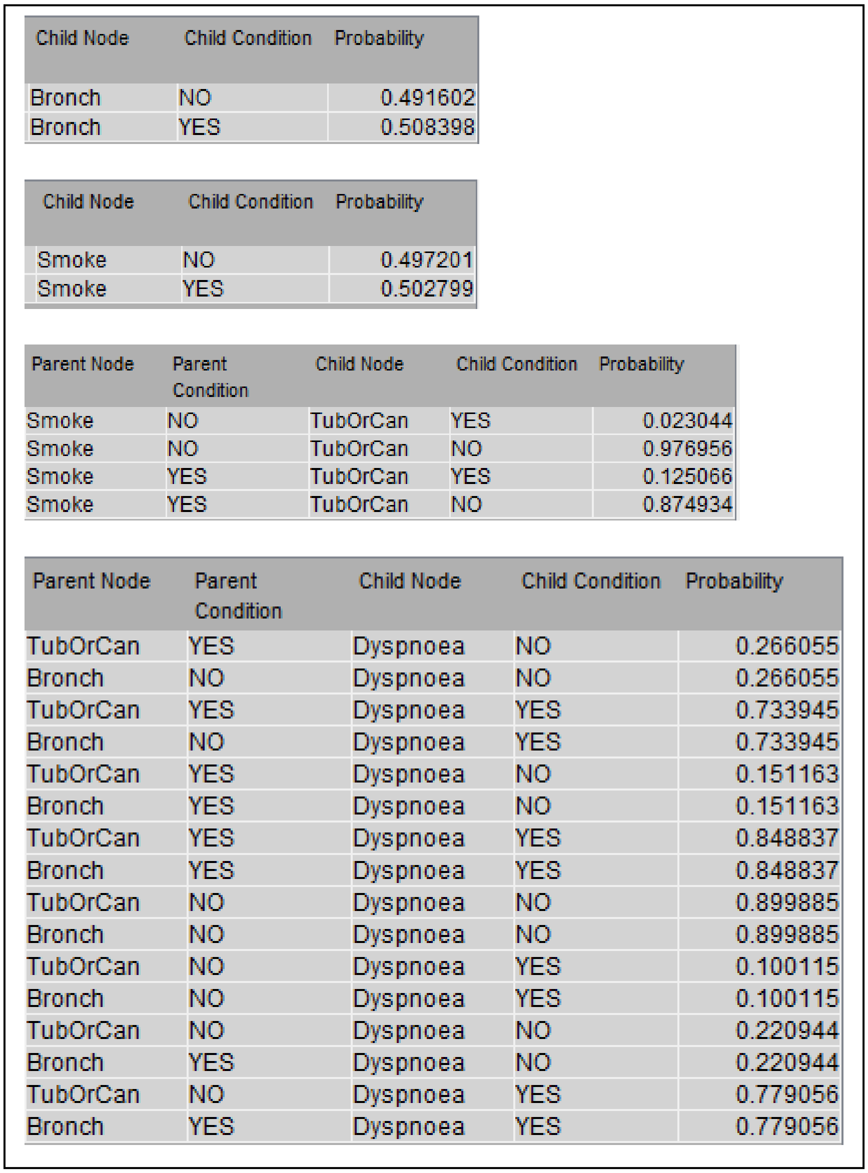
For example, from the table in Output 2, a patient who presents with shortness-of-breath but without bronchitis has a 0.315 probability of having tuberculosis or cancer. Stated another way, tuberculosis or cancer causes shortness of breath with probability 0.315.

The fitted BN-PC structure differs from the "true" structure in Lauritzen and Spiegelhalter (1988). The Markov blanket structure replicates their structure except that we omitted the individual indicators *Tuber* and *Cancer*, which are parents of

TubOrCan. They also show a directed arc from *Visit* into *Tuber*. Rather than choosing the BN-PC structure, which is strictly data-driven, we could choose the Markov blanket structure based on subject matter expertise that states that bronchitis is a probable cause of shortness of breath. That is, bronchitis should be a parent of dyspnea rather than a spouse, as in the PC structure.

The Markov blanket posterior probabilities for bronchitis, smoking, tuberculosis-or-cancer, and dyspnea, are shown in Output 3.

Output 3. Markov Blanket BN Posterior Probabilities



Child Node	Child Condition	Probability
Bronch	NO	0.491602
Bronch	YES	0.508398

Child Node	Child Condition	Probability
Smoke	NO	0.497201
Smoke	YES	0.502799

Parent Node	Parent Condition	Child Node	Child Condition	Probability
Smoke	NO	TubOrCan	YES	0.023044
Smoke	NO	TubOrCan	NO	0.976956
Smoke	YES	TubOrCan	YES	0.125066
Smoke	YES	TubOrCan	NO	0.874934

Parent Node	Parent Condition	Child Node	Child Condition	Probability
TubOrCan	YES	Dyspnoea	NO	0.266055
Bronch	NO	Dyspnoea	NO	0.266055
TubOrCan	YES	Dyspnoea	YES	0.733945
Bronch	NO	Dyspnoea	YES	0.733945
TubOrCan	YES	Dyspnoea	NO	0.151163
Bronch	YES	Dyspnoea	NO	0.151163
TubOrCan	YES	Dyspnoea	YES	0.848837
Bronch	YES	Dyspnoea	YES	0.848837
TubOrCan	NO	Dyspnoea	NO	0.899885
Bronch	NO	Dyspnoea	NO	0.899885
TubOrCan	NO	Dyspnoea	YES	0.100115
Bronch	NO	Dyspnoea	YES	0.100115
TubOrCan	NO	Dyspnoea	NO	0.220944
Bronch	YES	Dyspnoea	NO	0.220944
TubOrCan	NO	Dyspnoea	YES	0.779056
Bronch	YES	Dyspnoea	YES	0.779056

You can conduct scenario analyses to assess the impact of an intervention by fixing the value of the corresponding node. For example, to determine if bronchitis is a cause of dyspnea, the variable *Bronch* can be set to “yes”; i.e. P(*Bronch* = yes) =1. To assess causality for dyspnea, we calculate: P(Dyspnea = yes | *Bronch* = yes). Because *Dyspnea* also has *TubOrCan* as a parent, we must calculate this probability over both values of *TubOrCan*. I.e.

P(*Dyspnea* = yes | *Bronch* = yes, *TubOrCan* = yes) x P(*TubOrCan* = yes) + P(*Dyspnea* = yes | *Bronch* = yes, *TubOrCan* = no) x P(*TubOrCan* = no). Because *TubOrCan* also has a parent, Smoke, its probabilities must be calculated by summing over values of *Smoke*. The complete calculation is:

P(*Dyspnea* = yes | *Bronch* = yes) =

.85x(.02x.5 + .13x.5) + .78x(.98x.5 + .87x.5) = 0.79.

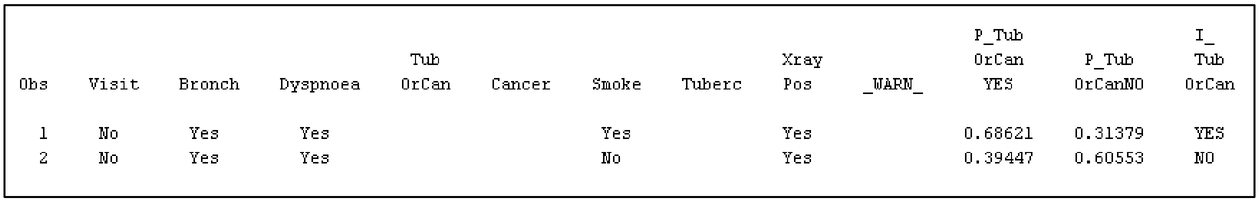
Similarly, we calculate the probability of Dyspnea in the absence of bronchitis;

.73x(.02x.5 + .13x.5) + .10x(.98x.5 + .87x.5) = 0.15.

To determine the effect of bronchitis on the shortness-of-breath, we can calculate the difference, 0.79 – 0.15 = 0.64, or risk ratio, 0.79/0.15 = 5.27. We can conclude that Bronchitis is a positive cause of Dyspnea. Of course, because the probability tables are stored as data sets, you can automate calculations of this sort if they are known to be of interest in advance.

It is also possible to score a data set of observations to obtain predictions of interest. For example, we created two observations with input values as shown in Output 4. The difference between the two is the value of smoking; one is set to “yes” and the other “no’’. The predicted probability for the outcome, *TubOrCan*, is stored in the variable *P_TubOrCanYES*. Because HPBN is a classifier, EM will automatically assign each observation to an outcome class with the higher predicted probability.

Output 4 Two Observations Scored Using the BN Markov Blanket



```
                                                                                          P_Tub                 I_
                                               Tub                              Xray      OrCan      P_Tub      Tub
Obs    Visit    Bronch    Dyspnoea    OrCan    Cancer    Smoke    Tuberc    Pos    _WARN_    YES      OrCanNO    OrCan

  1     No       Yes        Yes                           Yes                Yes              0.68621    0.31379    YES
  2     No       Yes        Yes                           No                 Yes              0.39447    0.60553    NO
```

CONCLUSION

This paper describes Bayesian networks (BN), the construction of BNs in SAS®, and how to use BNs for causal inference. An example based on the *Asia* data set is given by an implementation in SAS® Enterprise Miner using the HPBN classifier node.

In cases with continuous variables, SAS® Enterprise Miner bins continuous variables into equal-width levels and treats them as categorical variables. The default number of levels is 5, and users can manually specify the number of levels.

BN learning methods described in this paper are assumed to be based on observational and experimental data where observations in a sample are assumed to be collected under the same general conditions. In cases with interventional data, values of specific variables (interventions) are set by an external intervention. BN structures can be learnt from a structure in which the intervention variables are included and all other variables are depended on them. It can be done by adding specific arcs from intervention variables into other variables, which could be realized in the future release of HPBNET node in SAS® Enterprise Miner.

Real-world variables or factors comprise complex systems which may evolve over time. The set of values of the variables or factors are a time-stamped state and, at any point in time, there is a prior state, transition, and current state. Dynamic Bayesian networks can be used to model dependencies between factors of interests arising from such time series. Examples include system reliability models and models of operational risk in finance.

REFERENCES

Amrhein, J., Wang, F. 2018. "Bayesian Concepts: An Introduction". Proceedings of SAS Global Forum 2018, Paper 1863-2018. Denver, Colorado. SAS Institute, Cary NC.

Liu, Y., Shi, W., and Czika, W. 2017. "Building Bayesian Network Classifiers Using the HPBNET Procedure." Proceedings of SAS Global Forum 2017, Paper 474-2017. Orlando, FL. SAS Institute, Cary NC.

Pearl, J. 2009. *Causality: Models, Reasoning, and Inference*. 2nd ed. Cambridge: Cambridge University Press

Pearl, J., Glymour, M., Jewell, N. 2016. *Causal Inference in Statistics*. A Primer. Wiley

Lautitzen, S.L. and Spiegelhalter, D.J. 1988. "A Local Computations with Probabilities on Graphical Structures and Their Applications to Expert Systems." Journal of the Royal Statistical Society. Series B (Methodological), Vol. 50, No. 2, 157-224.

ACKNOWLEDGMENTS

We would like to thank the Statistics Team in McDougall Scientific Ltd. for their support and advice.

RECOMMENDED READING

- *SAS® Enterprise Miner™ 14.3 Guide*

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Fei Wang
McDougall Scientific Ltd.
fwang@mcdougallscientific.com
http://www.mcdougallscientific.com

Paper 3188-2019

The Bayesians are Coming! The Bayesians are Coming! The Bayesians are Coming to Time Series!

Aric LaBarr, Institute for Advanced Analytics at North Carolina State University

ABSTRACT

With the computational advances over the past few decades, Bayesian analysis approaches are starting to be fully appreciated. Forecasting and time series also have Bayesian approaches and techniques, but most people are unfamiliar with them due to the immense popularity of Exponential Smoothing and ARIMA classes of models. However, Bayesian modeling and time series analysis have a lot in common! Both are based on using historical information to help inform future modeling and decisions. This talk will compare the classical Exponential Smoothing and ARIMA class models to Bayesian models with autoregressive components. It will compare results from each of the classes of models on the same data set as well as discuss how to approach Bayesian time series models in SAS.

INTRODUCTION

Forecasting has applications across all industries. From needing to predict future values of sales for a product line, energy usage for a power company, to volatility of a portfolio of assets to hedge against risk, forecasting provides needed answers to decision makers. Time series models are extensively used in industry for this very purpose. Time series models and the underlying data are different in design than traditional cross-sectional modeling techniques. Typical cross-sectional modeling techniques are calculated across many individuals or objects usually under some assumption of independence. Whether they are traditional statistical models or newer machine learning algorithms, the techniques ignore, or are calculated regardless of, any time component. A simple example of this is analyzing energy usage across buildings on a major university campus compared to the temperature outside to understand the relationship between the two. Time series models on the other hand use the evolutionary nature of a set of data across time to help with the forecasting of future values. Their data typically consists of series of observations measured on the same individual or object across many points in time. A similar example for time series modeling is following the progress of energy usage for a building across time and use the previous values of energy usage to forecast future ones.

Popular approaches to time series forecasting are exponential smoothing models and combinations/variations on autoregressive and moving average models called ARIMA models. Both of these families of approaches are typically done with frequentist statistical methodology compared to Bayesian statistical methodology. Bayesian statistical methodology is named after Thomas Bayes, an English statistician, philosopher, and Presbyterian minister. The overly simplified difference between Bayesian and frequentist statistical methodology revolves around the estimation of unknown population parameters such as coefficients in a regression model. To a frequentist, these coefficients are fixed quantities that we are trying to estimate with a single point. Variability exists in these point estimates, but they are still single point estimates. To a Bayesian, these coefficients are random variables that follow a probability distribution, so they estimate entire distributions instead of single points. To do this, Bayesians make assumptions on the prior state of coefficients to inform their future thought on the coefficients' distributions. Each are making assumptions on the parameters of interest, the frequentist assuming the parameter doesn't move, and the Bayesian assuming that not only does the parameter move, but they know how the parameter moves.

Theoretical arguments aside, practitioners benefit from having knowledge of both frequentist and Bayesian time series modeling approaches. The more techniques a practitioner has, the better chance they have at providing the best solution to the decision maker using the forecasts, which is the true end goal. This paper will review the traditional frequentist approaches to time series – exponential smoothing models, ARIMA models, and VAR models. Next, the paper will introduce Bayesian time series approaches – Bayesian autoregressive models and Bayesian VAR models – hoping to build out the readers tools for forecasting real world problems using SAS® software.

CLASSICAL TIME SERIES MODELING

At its heart, time series analysis basically tries to break down a series of data into two primary components – signal and noise. We extract the signal from the data and repeat this signal into the future while using the noise to estimate variation in our signal. Specifically, in time series we rely on the assumption that the observations at a certain point in time depend on previous observations in time. There are two extremes to this approach. The first is the naïve model which depends only on the previous observation,

$$\hat{Y}_{t+h} = Y_t$$

where Y_t is the last observed data point, and $\hat{Y}_{t+h}$ is the forecast going h observations into the future. The second approach is the overall average that weights each observation in time equally:

$$\hat{Y}_{t+h} = \frac{1}{T}\sum_{t=1}^{T} Y_t$$

Exponential smoothing models as well as ARIMA class models are a balance between these two extremes. Throughout this paper, we will try to forecast the percentage change in quarterly United States personal consumption expenditure (PCE) – essentially household buying habits based on the price changes in consumer goods and services. Figure 1 displays

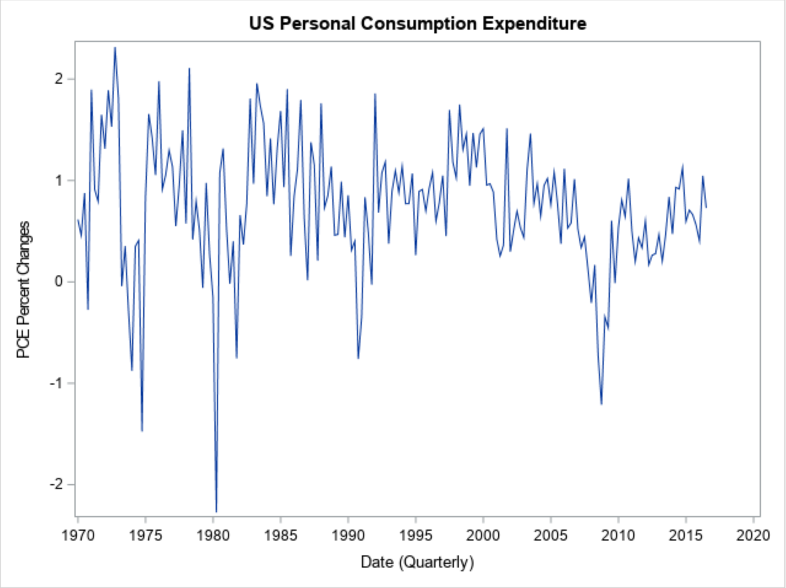


Figure 1: US Personal Consumption Expenditures

the quarterly US PCE from Q1 1970 through Q3 2016. For all of the analysis in this paper, the last seven observations – 2015 and 2016 – were removed and used as a hold-out sample for model accuracy reporting.

EXPONENTIAL SMOOTHING MODELS

Exponential smoothing models (ESMs) became popular after WWII because of their accurate forecasting with minimal hand calculation. They are still popular today because those same principles are important to business modeling today. ESMs emphasize more recent observations in their weighting scheme for forecasting observations. There are many different types and classes of ESMs, but this paper uses the simple exponential smoothing model that doesn't include trend or seasonality as our data doesn't exhibit either. The simple (or single) ESM applies a weighting scheme on observations that decreases exponentially the further back in time we go,

$$\hat{Y}_{t+1} = \theta Y_t + \theta(1-\theta)Y_{t-1} + \theta(1-\theta)^2 Y_{t-2} + \theta(1-\theta)^3 Y_{t-3} + \cdots$$

where θ is bounded between 0 and 1. The larger the value of θ, the more that the most recent observation is emphasized as seen in Figure 2.

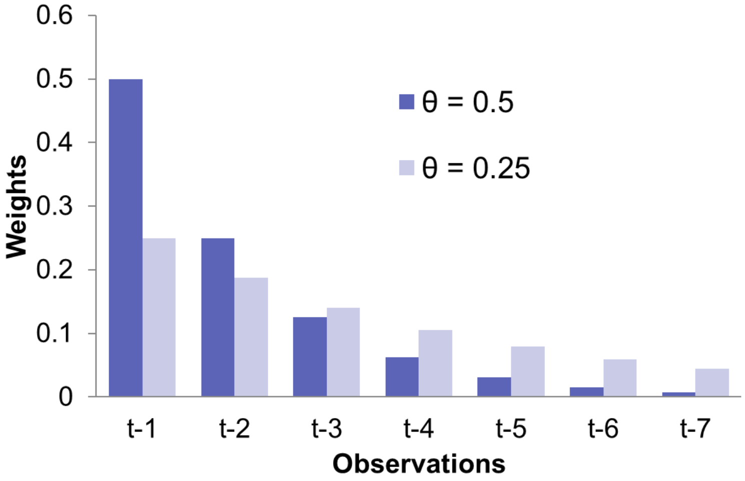


Figure 2: Comparing Weights Across Theta Values

The above exponentially decreasing weights simplify to the following equation:

$$\hat{Y}_{t+1} = \theta Y_t + (1-\theta)\hat{Y}_t$$

These models minimize the sum of the squared errors on the one-step ahead forecasts, essentially optimizing themselves to forecast one time period into the future.

The following SAS procedure computes the simple ESM:

```
proc esm data = train print = all plot = all lead = 7 outfor = for_esm;
  forecast consumption / model = simple;
run;
```

The optimized value of $\theta = 0.334$. The forecasts of a simple ESM are rather boring as they are a horizontal line at the forecast for the next time period. Since the simple ESM is optimized to predicting one time period into the future, it uses that one forecast infinitely into the future as seen in the forecast charts in Figure 3.

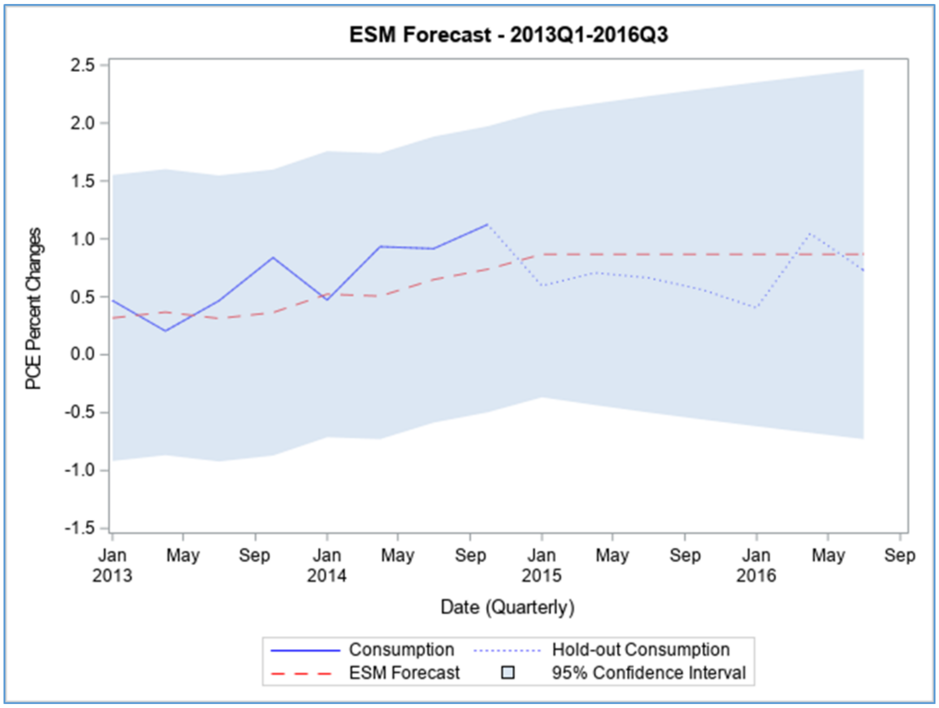


Figure 3: ESM Forecast for US Personal Consumption Expenditure

The code to subset the needed data and produce the previous figure is the following:

```
data esm_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_esm;
run;

proc sgplot data=esm_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'ESM Forecast - 2013Q1-2016Q3';
  series x = Index y = ACTUAL / lineattrs = (color = blue) name = "train"
    legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = PREDICT / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "ESM Forecast";
  band x = Index upper=UPPER lower=LOWER / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

ESMs predict fairly well since that is their design, however, they may still not find all of the signal in a series.

ARIMA MODELS

Exponential smoothing models limit the structure on how previous observations impact future forecasts to only exponentially decreasing. There is no guarantee that that specific structure must hold true. Autoregressive (AR) models forecast a series of observations based solely on past values. The main difference from ESMs is that AR models allow each lag, or previous observation, to have values not necessarily decreasing exponentially the further back in time you go. In fact, there may come a point when previous lags values of a series of data no longer impacts the current observation and the AR model stops at that lag and doesn't continue going further back in time. AR models are long memory models where the effects of observations dissipate over many time periods.

The following is the AR model that has p lags:

$$\hat{Y}_t = \hat{\alpha}_0 + \hat{\alpha}_1 Y_{t-1} + \hat{\alpha}_2 Y_{t-2} + \cdots + \hat{\alpha}_p Y_{t-p}$$

To help mitigate large values of p, combining AR models with moving average (MA) models to form ARIMA models can reduce the parameterization of a model. MA models are short memory models that forecast future values based solely on errors from previous time points. The MA model with q previous error terms is the following:

$$\hat{Y}_t = \hat{\alpha}_0 + \hat{\beta}_1 \varepsilon_{t-1} + \hat{\beta}_2 \varepsilon_{t-2} + \cdots + \hat{\beta}_q \varepsilon_{t-q}$$

Combining these models together form the ARIMA model. The I in ARIMA stands for integrated and is not covered here in this paper as it deals with making the data stationary, which our example data is.

Three available model selection techniques in SAS are the minimum information criterion (MINIC), the smallest canonical correlation (SCAN), and the extended sample autocorrelation function (ESACF). The technical details for each of these is not provided here in this paper, but each provides an estimate of the number of AR and MA lags needed to best fit the data. The following code runs these selection techniques looking up to 4 lags into the past:

```
proc arima data=train plot=all;
  identify var=consumption nlag=10 minic scan esacf P=(0:4) Q=(0:4);
run;
quit;
```

Two of the techniques – the MINIC and SCAN – agree that the AR(3) model with no MA terms is the best model:

$$\hat{Y}_t = \hat{\alpha}_0 + \hat{\alpha}_1 Y_{t-1} + \hat{\alpha}_2 Y_{t-2} + \hat{\alpha}_3 Y_{t-3}$$

The following code builds the AR(3) model and forecasts seven observations forward:

```
proc arima data=train plot=all;
  identify var=consumption nlag=10;
  estimate p=3 method=ML;
  forecast lead=7 out=for_arima;
run;
quit;
```

The forecast by the AR(3) model suggests a slight downward projection for future values of PCE as seen in Figure 4.

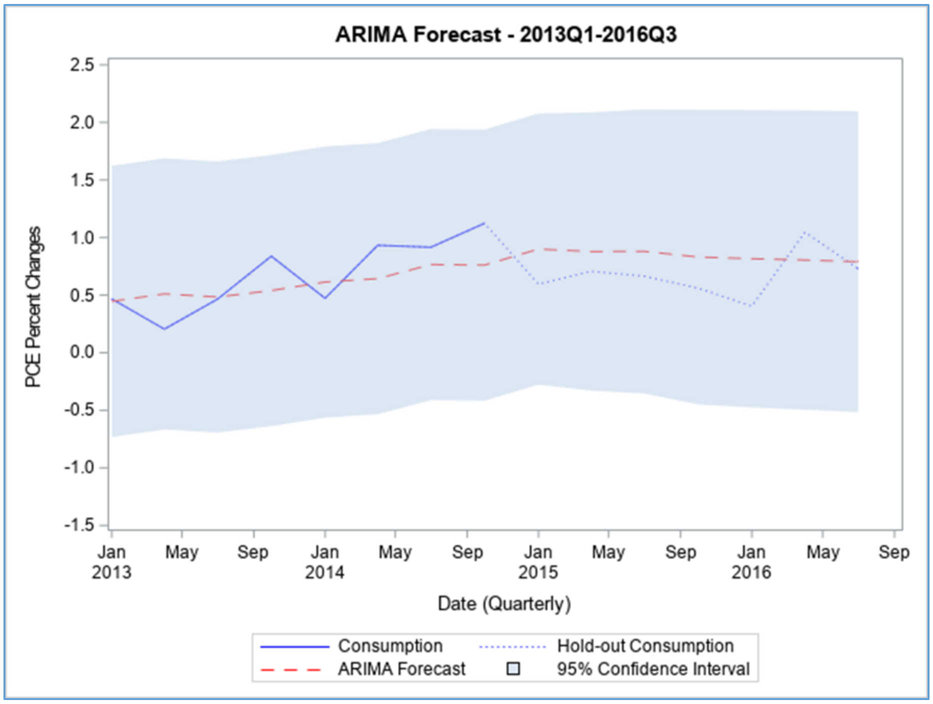


Figure 4: ARIMA Forecast for US Personal Consumption Expenditure

The code to subset the needed data and produce the previous figure is the following:

```
data arima_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_arima;
run;

proc sgplot data=arima_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'ARIMA Forecast - 2013Q1-2016Q3';
  series x = Index y = Consumption / lineattrs = (color = blue)
    name = "train" legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = Forecast / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "ARIMA Forecast";
  band x = Index upper=U95 lower=L95 / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

Both ARIMA models and ESMs are univariate classes of models, predicting only one series at a time. However, what if PCE moves with changes in personal income? Forecasting both of these simultaneously might provide further information.

VECTOR AUTOREGRESSIVE MODELS

Multivariate analysis tries to predict many different response variables at the same time in the same model. Instead of single variables, we use vectors of variables. Instead of single coefficients, matrices of coefficients are used. An example of a vector AR(1) model looks like the following:

$$\begin{bmatrix} Y_{t,1} \\ Y_{t,2} \end{bmatrix} = \begin{bmatrix} \alpha_{0,1} \\ \alpha_{0,2} \end{bmatrix} + \begin{bmatrix} \alpha_{11,1} & \alpha_{12,1} \\ \alpha_{21,1} & \alpha_{22,1} \end{bmatrix} \begin{bmatrix} Y_{t-1,1} \\ Y_{t-1,2} \end{bmatrix} + \begin{bmatrix} e_{t,1} \\ e_{t,2} \end{bmatrix}$$

By expanding out the matrices the single vector equation becomes two equations, one for each of the response variables:

$$Y_{t,1} = \alpha_{0,1} + \alpha_{11,1}Y_{t-1,1} + \alpha_{12,1}Y_{t-1,2} + e_{t,1}$$

$$Y_{t,2} = \alpha_{0,2} + \alpha_{21,1}Y_{t-1,1} + \alpha_{22,1}Y_{t-1,2} + e_{t,2}$$

The benefit may be better seen through this view of the models as it shows that each response variable depends not only on its previous lagged values, but also the lagged values of the other response variable. In our data set, this means that previous values of changes in PCE as well as previous values of changes in personal income impact the forecasts of PCE going forward. Due to our previous analysis using an AR(3) model for predicting univariate PCE, we use a VAR(3) model with percentage changes in personal income as the other response variable. The VAR(3) model is defined as follows:

$$\boldsymbol{\widehat{Y}_t = \widehat{\alpha}_0 + \widehat{\alpha}_1 Y_{t-1} + \widehat{\alpha}_2 Y_{t-2} + \widehat{\alpha}_3 Y_{t-3}}$$

There is a different SAS procedure to deal with multivariate time series as compared to univariate time series:

```
proc varmax data = train plot=all;
  id index interval=quarter;
  model consumption income / p=3 lagmax=5 print=(estimates diagnose);
  output out=for_var lead=7;
run;
```

Similar to the AR(3) model, the VAR(3) model also forecasts a slight downward trend over the next seven time periods as seen in Figure 5.

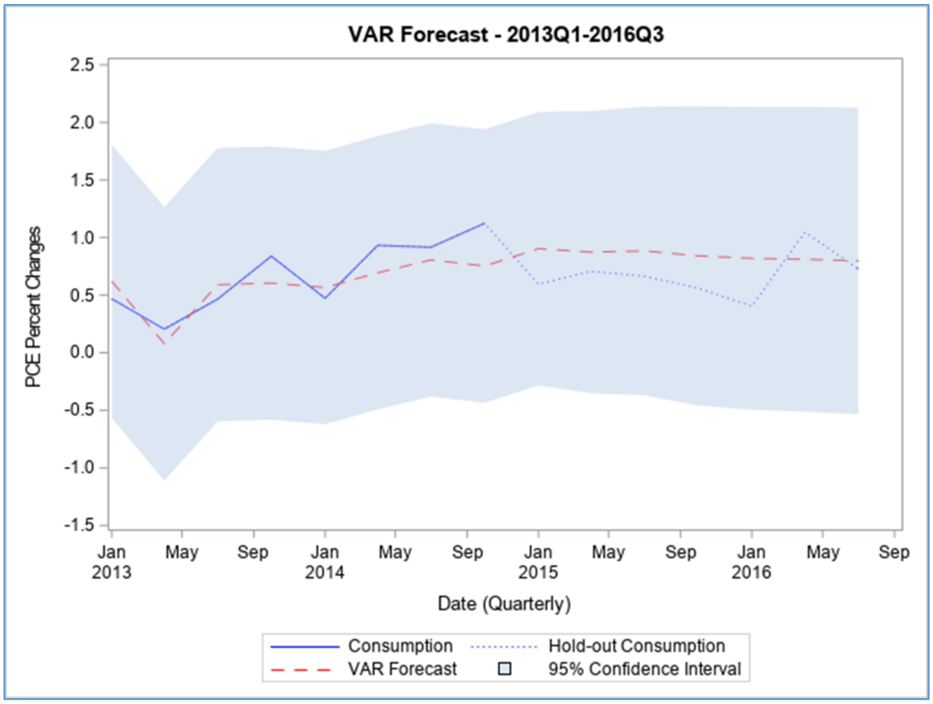


Figure 5: VAR Forecast for US Personal Consumption Expenditures

The code to subset the needed data and produce the previous figure is the following:

```
data var_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_var;
run;

proc sgplot data=var_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'VAR Forecast - 2013Q1-2016Q3';
  series x = Index y = Consumption / lineattrs = (color = blue)
    name = "train" legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = For1 / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "VAR Forecast";
  band x = Index upper=UCI1 lower=LCI1 / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

This section has served as a brief review of popular time series models. All of the previous models take a frequentist approach to modeling time series. Time series models inherently depend on previous knowledge using lagged variables. The Bayesian framework of modeling relies on previous assumptions about data, which fits in perfectly with time series.

BAYESIAN TIME SERIES MODELING

The wonderful part about Bayesian time series modeling is that the structures of the models are mostly identical to frequentist models. The main difference is the assumptions. Instead of just taking the inputs into our model – the previous data and parameters – as fixed values we are either using or estimating, we assume that they are random variables with corresponding distributions.

BAYESIAN AUTOREGRESSIVE MODELS

We are going to fit a Bayesian AR(3) model since the AR(3) structure seemed to fit our data the best according to model selection techniques. Estimating the distribution of input variables, here the lagged values of consumption, is rather easy to do. Since the residuals

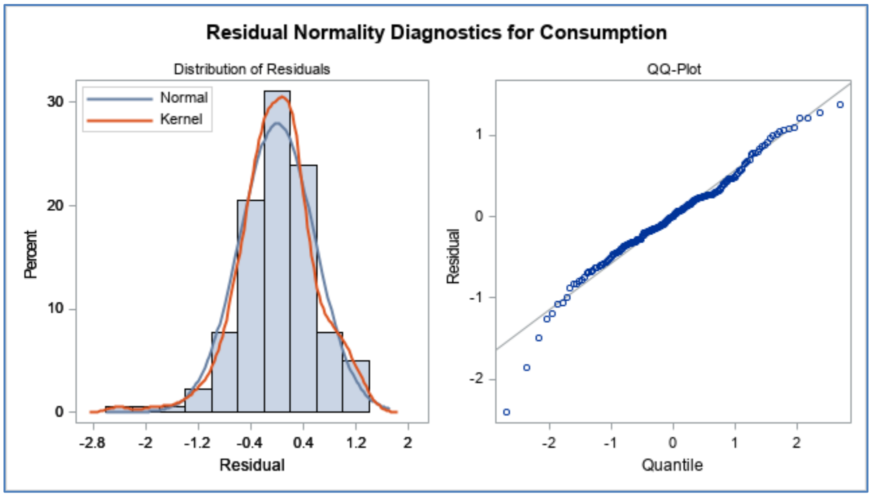


Figure 6: Residuals from AR(3) Model

from our original AR(3) model are approximately normal as seen in Figure 6, with only a little difference occurring with some outliers in the tail, we will assume the consumption model also follows a normal distribution with a mean of the AR(3) estimated AR(3) model and variance of σ^2. However, we must also assume distributions on all of the parameters from the AR(3) model – $\alpha_1, \alpha_2, \alpha_3, \sigma^2$. Table 1 lists all of the assumed distributions for the model.

Random Variables	**Assumed Distribution**
Consumption	Normal Mean = AR(3), S.D. = σ^2
Consumption Initial Value 0	Normal Mean = 0, S.D. = 1
Consumption Initial Value 1	Normal Mean = 0, S.D. = 1
Consumption Initial Value 2	Normal Mean = 0, S.D. = 1
α_1 – AR(1) Parameter	Normal Mean = 0, S.D. = 1
α_2 – AR(2) Parameter	Normal Mean = 0, S.D. = 1
α_3 – AR(3) Parameter	Normal Mean = 0, S.D. = 1
σ^2 – Model Variance	Gamma Shape = 3/10, Scale = 10/3

Table 1: Prior Distributions in Bayesian AR(3) Model

In Bayesian times series analysis, we estimated the posterior distributions of the final forecasts through Markov Chain Monte Carlo (MCMC) techniques. Specifically, PROC MCMC in SAS uses the random walk Metropolis algorithm to calculate the posterior distribution. The mathematical details of this algorithm is not covered in this paper, but the general idea is that we want to simulate stepping through a series of probabilistic events connected to one another. If we draw enough samples from this series of events, we will eventually get a distribution that resembles the posterior distribution that we desire – forecasted future values. For each forecasted future value we actually have a distribution of possible future values at that time point. We take the mean of the distribution of possible values at each of the seven forecasted time points developing our forecasts over the next seven quarters. SAS easily does this with PROC MCMC:

```
proc mcmc data = train nmc = 100000 seed = 100 nthreads = 8 propcov=quanew;
  parms alpha_1 alpha_2 alpha_3;
  parms sigma2 1;
  parms Y_0 Y_1 Y_2;

  prior alpha_: ~ normal(0,var = 1);
  prior sigma2 ~ igamma(shape = 3/10, scale = 10/3);
  prior Y_: ~ normal(0, var = 1 );

  mu = alpha_1*consumption.l1 + alpha_2*consumption.l2 +
       alpha_3*consumption.l3;
  model consumption ~ normal(mu, var = sigma2) icond=(Y_2 Y_1 Y_0);
```

```
    preddist outpred=predicted statistics=brief;

    ods output PredSumInt=for_bar;
  run;
```

with the NMC option specifying the number of MCMC iterations. The remaining pieces of the code define the prior distributions and the form of the model to be forecasted. The variables consumption.l1, consumption.l2, and consumption.l3 are lag values created by PROC MCMC, which makes the managing of data much easier.

The Bayesian AR(3) model seems to have the closest fit yet of the forecasted models for consumption as seen in Figure 7.

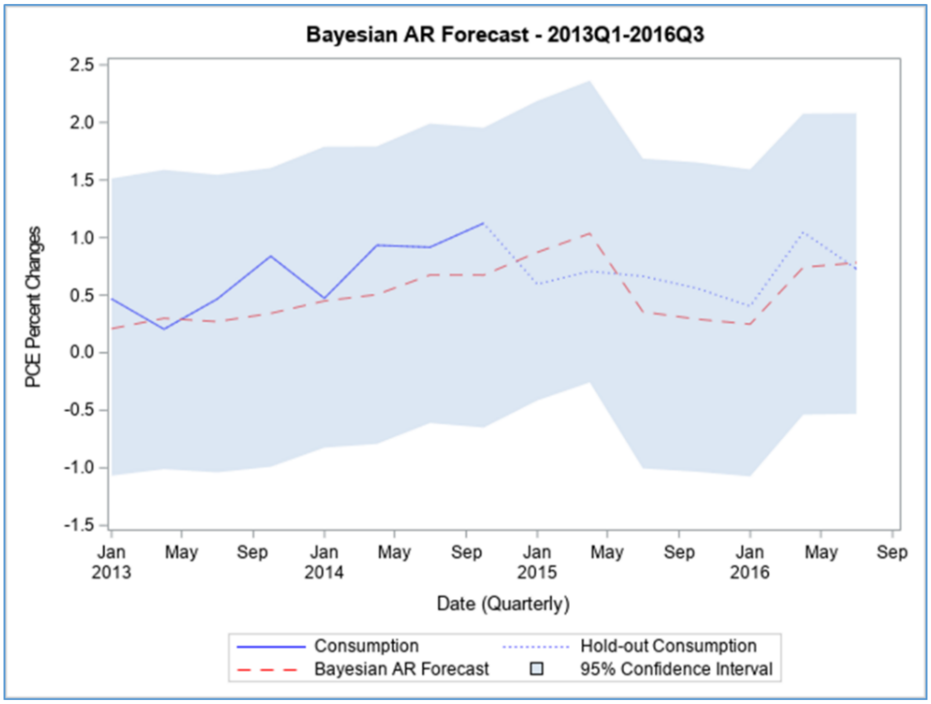


Figure 7: Bayesian AR(3) Forecast of US Personal Consumption Expenditures

The code to subset the needed data and produce the previous figure is the following:

```
  data bar_chart;
    set us_econ(rename=(Consumption=Full_Consumption));
    set train;
    set for_bar;
  run;

  proc sgplot data=bar_chart(where=('01JAN2013'd <= Index)) noautolegend;
    title 'Bayesian AR Forecast - 2013Q1-2016Q3';
    series x = Index y = Consumption / lineattrs = (color = blue)
      name = "train" legendlabel = "Consumption";
    series x = Index y = Full_Consumption / lineattrs = (color = blue
      pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
    series x = Index y = Mean / lineattrs = (color = red pattern = dash)
      name = "for" legendlabel = "Bayesian AR Forecast";
    band x = Index upper=hpdupper lower=hpdlower / transparency=.5
```

```
      name = "CI" legendlabel = "95% Confidence Interval";
    yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
    xaxis label='Date (Quarterly)';
    keylegend "train" "test" "for" "CI";
  run;
```

If frequentist AR models can be extended into the multivariate space, we can do the same thing with Bayesian AR models as well. Again, we will try to forecast both changes in PCE and personal income simultaneously to try and improve our model.

BAYESIAN VECTOR AUTOREGRESSIVE MODELS

Similar to the extension of the Bayesian AR model to the frequentist AR model framework, the Bayesian extension to vector AR models is rather straight-forward. The model structure for the Bayesian VAR model is the same as the frequentist VAR model, but with additional assumptions on the parameters and known lagged values of the data. In PROC VARMAX, the multivariate normal distribution is assumed for the needed data and parameters. The only options to control the fit of the distribution are the hyperparameters in these Bayesian VAR models with trial and error being the best approach.

Similar to the previous models, three lagged values are used in the model – a Bayesian VAR(3):

```
  proc varmax data = train plot=all;
    id index interval=quarter;
    model consumption income / p=3 lagmax=5
          print=(estimates diagnose)
          prior=(lambda=0.9 theta=0.1);
    output out=for_bvar lead=7;
  run;
```

The Bayesian VAR(3) model is very similar in forecasts as the VAR(3) model as seen in Figure 8. The univariate models seemed to improve with the addition of the Bayesian framework, but the multivariate models seemed to remain approximately the same.

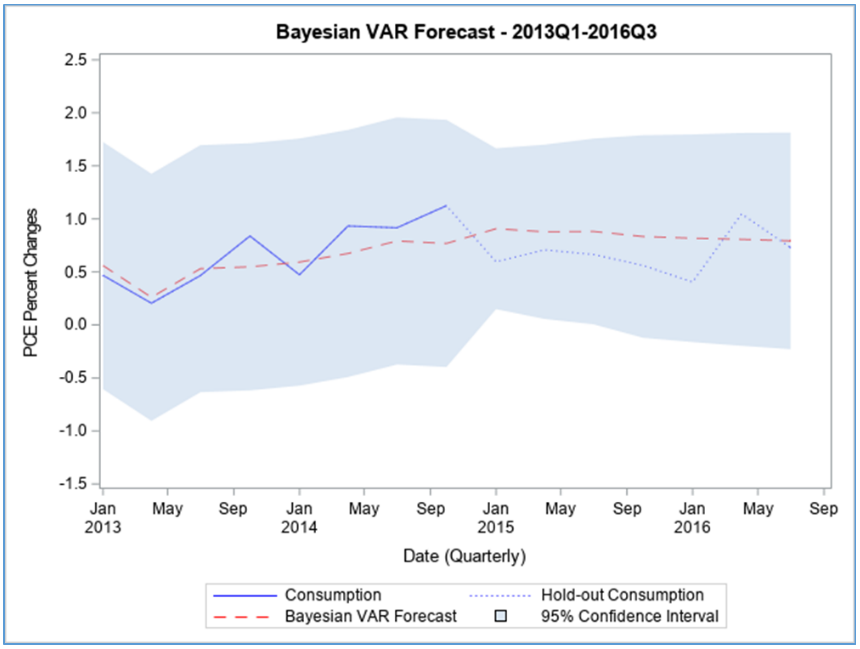


Figure 8: BVAR(3) Forecast of US Personal Consumption Expenditures

The code to subset the needed data and produce the previous figure is the following:

```
data var_chart;
  set us_econ(rename=(Consumption=Full_Consumption));
  set for_var;
run;

proc sgplot data=var_chart(where=('01JAN2013'd <= Index)) noautolegend;
  title 'VAR Forecast - 2013Q1-2016Q3';
  series x = Index y = Consumption / lineattrs = (color = blue)
    name = "train" legendlabel = "Consumption";
  series x = Index y = Full_Consumption / lineattrs = (color = blue
    pattern = dot) name = "test" legendlabel = "Hold-out Consumption";
  series x = Index y = For1 / lineattrs = (color = red pattern = dash)
    name = "for" legendlabel = "VAR Forecast";
  band x = Index upper=UCI1 lower=LCI1 / transparency=.5 name = "CI"
    legendlabel = "95% Confidence Interval";
  yaxis label='PCE Percent Changes' values=(-1.5 to 2.5 by 0.5);
  xaxis label='Date (Quarterly)';
  keylegend "train" "test" "for" "CI";
run;
```

Each of the models had varying forecasts for the future seven quarters of US PCE. Combining some of the forecasts into an ensemble might improve them even further.

CONCLUSION

This paper summarized and introduced some basic tools that every forecasting practitioner should know. In each problem, it is best to compare across modeling techniques in a hold-out sample to know which one is best to use. Instead of just visualizing the results from above, we calculated the Mean Absolute Percentage Error for each forecast on the hold-out data set:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right|$$

Table 2 contains the MAPE values for each of the models proposed above.

Model	Mean Absolute Percentage Error
Bayesian Autoregressive(3)	37.4%
Autoregressive(3)	41.3%
Bayesian Vector Autoregressive(3)	41.6%
Vector Autoregressive(3)	41.8%
Simple Exponential Smoothing Model	43.4%

Table 2: MAPE Values From Proposed Models

The univariate Bayesian AR(3) model performed the best when compared using MAPE on the hold-out sample of seven quarters. One last tool that any practitioner in forecasting should have is the technique of ensembling forecasts together. Visual inspection of both Figure 4 and Figure 7 show that our two top performing models predict in opposite directions. The AR(3) model overpredicts change in PCE, while the Bayesian AR(3) underpredicts it. Averaging these forecasts together proves to outperform either of the forecasts individually. The MAPE from this ensemble of forecasts – straight average of the two – is **22.4%**. This is a definite improvement easily seen in Figure 9.

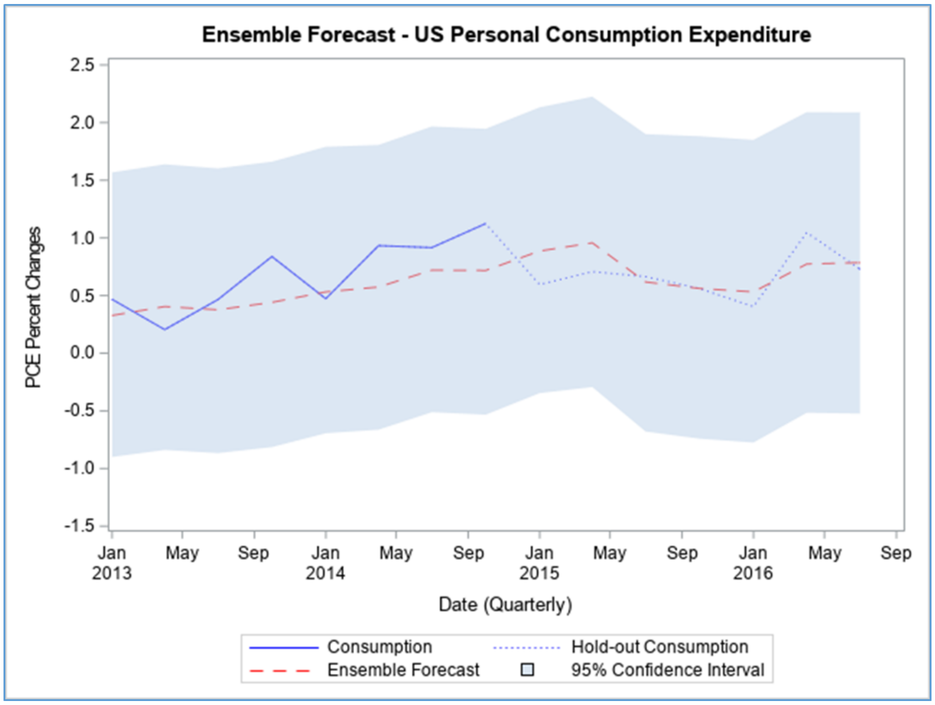


Figure 9: Ensemble Forecast of US Personal Consumption Expenditures

Both frequentist and Bayesian approaches to time series are valuable tools for any practitioner in forecasting. Each of the approaches contains their own theoretical foundation, which we will not argue for one way or the other as each forecasting problem is unique and there are no clear favorites to approaches. Every approach is a viable one for forecasting problems that arise, especially when you can combine them through ensembling of forecasts. Only the practitioner of the problem would know best. Hopefully now the practitioner has more tools in their tool belt!

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Aric LaBarr
Institute for Advanced Analytics at North Carolina State University
919-513-4076
aric_labarr@ncsu.edu
http://www.ariclabarr.com

SAS® GLOBAL FORUM 2020 #SASGF

Paper 4311-2020

Incorporating Auxiliary Information into Your Model Using Bayesian Methods in SAS® Econometrics

Matthew Simpson, SAS Institute Inc.

ABSTRACT

In addition to data, analysts often have available useful auxiliary information about inputs into their model—for example, knowledge that high prices typically decrease demand or that sunny weather increases outdoor mall foot traffic. If used and incorporated correctly into the analysis, the auxiliary information can significantly improve the quality of the analysis. But this information is often ignored. Bayesian analysis provides a principled means of incorporating this information into the model through the prior distribution, but it does not provide a road map for translating auxiliary information into a useful prior. This paper reviews the basics of Bayesian analysis and provides a framework for turning auxiliary information into prior distributions for parameters in your model by using SAS® Econometrics software. It discusses common pitfalls and gives several examples of how to use the framework.

INTRODUCTION

Modern statistical analysis excels at generating insights from data, but these tools can take into account only the inputs that you provide them with. Often you have important information about the problem in the form of vague intuitions, but not in a quantifiable form that you can plug directly into your model. Further, even when you do have concrete auxiliary data relevant to your problem, it might not be straightforward to combine those data with your original data in a larger model. Bayesian analysis is an incredibly powerful means of taking into account various forms of auxiliary information through the so-called prior distribution. However, it is not straightforward to construct this prior, and a poorly constructed prior can yield significantly worse inferences than the ones you make by disregarding the auxiliary information altogether.

This paper provides a conceptual framework for thinking about the prior in order to make it easier for you to construct custom priors by using various sources of auxiliary information. The key is to transform the model and data in a variety of ways in order to make it easier to think about the model parameters and what they imply about observables. The general workflow is as follows: 1) convert the model and data to something that is easy for the analyst to have intuitions about; 2) convert those intuitions to numbers; and 3) convert those numbers to a prior distribution on the transformed version of the problem. To facilitate this process, the paper presents a number of rules of thumb for transforming the model and data into objects that are easier to think about and for converting intuitions into prior distributions.

The rest of the paper is organized as follows. The section “The Bayesian Story and its Discontents” sketches the subjective Bayesian philosophy of statistics and the problems with naively applying it, especially problems associated with the prior distribution. Then the section “Rules to Derive By” discusses some rules of thumb for overcoming these problems and translating various sorts of auxiliary information into prior distributions. Next “Example 1” introduces an example in order to illustrate how to apply the rules in a regression model. “Example 2” then introduces a new data source to illustrate how to apply the rules in probit regression. Continuing with the data in the original example, “Example 3” shows how to use the rules to construct an informative prior by incorporating the results of the probit regression, this time in the context of a count regression. Finally, the “Discussion” section summarizes and reviews the ideas in the paper.

THE BAYESIAN STORY AND ITS DISCONTENTS

The subjective Bayesian philosophy of statistics starts by identifying epistemic uncertainty with probability. That is, your uncertainty about a set of propositions should follow the rules of probability. Then Bayes' rule provides a convenient way to revise beliefs in light of new information. In its simplest form, Bayes' rule is

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

In practice B is the observed data, denoted by $\mathcal{D}$, and A is the model parameters, denoted by $\boldsymbol{\theta}$. This leads to the usual form of Bayes' rule:

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

In this notation, $p(\cdot)$ is the probability density of the enclosed quantity, so $p(\mathcal{D}|\boldsymbol{\theta})$ is the probability density of the data given any model parameters (that is, the likelihood), and $p(\boldsymbol{\theta})$ is the probability density of just the model parameters (that is, the prior density). To a Bayesian analyst, the likelihood tells you how you should update your beliefs about the data that you expect to see once you observe the model parameter, whereas the prior represents your beliefs about the model parameter before you see any data. Combined, they tell you how you should update your beliefs about future observations conditional on the observations that you have already seen. Then the posterior density, $p(\boldsymbol{\theta}|\mathcal{D})$, represents your beliefs about the model parameters after seeing the data. As in any statistical analysis, any questions that the analyst wants to answer need to be converted to statements about the model parameters.

This approach yields two major benefits. First, in principle, it enables the analyst to take into account other sources of information that are not normally considered as part of the "data," including relevant prior beliefs. This should yield more reliable inferences, in theory. Second, it enables the analyst to rigorously make probability statements. For example, in a forecast, the analyst might be able to rigorously say, "The probability that we meet revenue targets in the fourth quarter, conditional particular data sources, a particular model, and a particular prior is about 0.64." Such posterior probabilities are usually more directly relevant to decisions than, for example, the result of a hypothesis test.

Criticisms of Bayesian inference often focus on the prior distribution. Where does it come from? In practice, it can seem completely made up. This criticism is not completely unfair, but Bayesians have a glib response: it comes from the same place as the likelihood. They're both made up, in the sense that the analyst chooses them on the basis of some combination of judgment, experience, and convenience rather than, for example, basing the choice on some set of undeniable axioms. Even formal model selection criteria require judgment to use—for example, which criterion should you use? But according to Bayesians, you *always* have prior information. How often do you check to see whether your regression estimates are "reasonable"? The problem is that it is not obvious how to translate this information into a mathematically precise prior distribution. The Bayesian story is silent on this question, because it assumes that the prior already exists. Similarly, classical inference is silent on how to choose your significance level, because the theories of hypothesis testing and confidence intervals assume that it already exists. Answers to how, precisely, to deal with these issues help define the various flavors of Bayesian statistics, and indeed, the various non-Bayesian approaches to statistical inference.

The glib Bayesian response is insightful once it is fully fleshed out, but it does not completely justify the use of Bayesian methods. Even if you believe the arguments that subjective uncertainty follows the rules of probability, this does not mean that your statistical methods must be Bayesian. It might not be worth the hassle to translate all your auxiliary information into a mathematically precise prior distribution, and because of errors introduced in this process, a non-Bayesian method might actually yield inferences closer to the "true Bayesian" inferences.

But taking into account more information is appealing, as is the ability to make probability statements about quantities of interest. Even if you do not believe the Bayesian story, you can reconceptualize the benefits of the prior distribution in terms of regularization, or the bias-variance trade-off. To make this work, it is key to figure out how to operationalize the prior—how to construct and use it in real-world situations. Constructing a prior is no different from constructing the likelihood in a lot of ways. It typically is not obvious what the "correct" likelihood is, but often you can settle on a "good enough" likelihood while acknowledging its limitations for many uses.

RULES TO DERIVE BY

The previous section is careful to mention that the probability statements that are the output of Bayesian analysis are *conditional* on a variety of things, not just the observed data. It is obvious, but also worth making explicit, that posterior probabilities are also conditional on the likelihood and the prior—if you change either, then the posterior probabilities change. Together, the likelihood and the prior form a *model of your uncertainty* about the problem. From the subjective Bayesian perspective, the model is both pieces simultaneously. They are in some sense inseparable, and in fact the prior often does not make much sense outside the context of the likelihood. But it is usually easier to understand and construct the likelihood portion of the model than the prior, so it is OK to start there. The remainder of this paper assumes that a suitable likelihood has already been chosen, but keep in mind that this is not necessarily always true, and indeed the distinction between prior and likelihood is ambiguous in many contexts. All this leads to the first rule of thumb that you can use to construct prior densities.

Rule 1 *To choose a prior for a parameter, first understand how the parameter affects the distribution of observables.*

What does the regression coefficient imply about the observed responses, compared to the regression coefficient for other covariates? What about the error variance? These questions are not always easy to answer in any precise form. But there are several tricks that you can use to build your intuition. The key in each case is to start with something you have strong intuitions or relevant auxiliary information about—typically the distribution of the data. Then you convert that information to relevant information about the parameters in question.

The most general trick is transformation. Sometimes the scale of the data or the parameters is not easy to think about, so you should transform them to a quantity that is easier to think about.

Rule 2 *Transform quantities in the model to make them easier to think about.*

In practice this rule takes many forms, depending on the quantities in question. A simple but powerful example is to focus on standard deviations and correlations instead of variances and covariances. Standard deviations are already in the same units as the data, and the 68/95/99 rule for normal distributions allows for easy interpretation. If a distribution is approximately normal, then approximately 68% of the distribution is within one standard deviation of the mean, approximately 95% is within two standard deviations, and approximately 99% is within three standard deviations. Similarly, correlations are easier to understand than covariances because they are unitless. A correlation of 0.5 means the same thing no matter the units of the variables, so you can safely abstract away from units when constructing a prior.

Rule 3 *Construct priors for standard deviations and correlations instead of variances and covariances.*

A covariance matrix larger than 2×2 should be handled differently because of the positive definiteness constraint. Typically, a good strategy is to write the prior in terms of standard deviations and the correlation matrix, but these details are beyond the scope of this paper.

It is also useful to transform the data, most often by centering and scaling; that is, subtract the mean of the variable and then divide by its standard deviation to standardize the variable.

Rule 4 *Center and scale continuous covariates, and when applicable the response, in order to make regression coefficients easier to interpret.*

This rule ensures that each regression coefficient can be interpreted on a similar scale: “A change of one standard deviation in this covariate should predict how much of a change in the response?” Technically this is “using the data twice,” since you are using it in both the prior and the likelihood. This makes many Bayesians uncomfortable, because the prior is supposed to be completely independent from the data that you are analyzing. Preventing yourself from using the data to select the prior will make you look more like a Bayesian—as if you are following the procedure of Bayesian inference. But in practice, using the data to help operationalize the prior can help you get closer to the correct Bayesian answer (conditional on the likelihood, your prior information, and so on). So although it is not ideal, standardizing covariates is so useful in constructing priors that it is typically worth it. In principle, you can center and scale with population values if available, such as from census data, to obtain the same benefits for prior construction without using the data twice. The main caveat here is that if your sample is not representative of the population you are trying to make inferences about, then centering and scaling by the sample means and standard deviations might do more harm than good—for example, if your sample is too small.

Centering and scaling does not make much sense for categorical variables. So instead of transforming them, it is helpful to construct a base case. For example, assume that an observation is in a particular set of categories and the continuous covariates are set to particular values. What do you expect the observables to look like? How do you expect them to change if you move the observation to a different category? This can be complicated, but an easier process is to assume that the mean for the base case is some intuitive value. The examples in this paper illustrate this.

Rule 5 *Do not transform categorical covariates. Instead, construct a base case and think about what you expect for that base case and for changes from the base case to different categories.*

The basic idea of a base case works for continuous variables as well, especially after centering and scaling. For example, in nonlinear models or models with interactions, the impact of a covariate can depend on the values of the other covariates.

Rule 6 *If the impact of a continuous covariate depends on the values of other covariates, think about the impact of that covariate in the context of an intuitive base case.*

The key to choosing a base case is to pick one that is easy to think about, though this often depends on the model.

Another case where transforming is often not helpful is log-log models. That is, if the response is the log of the response that you care about, and the covariate is the log of the covariate that you care about, then the regression coefficient has an easy interpretation as an elasticity: “A 1% change in the covariate predicts a β% change in the response.”

Rule 7 *In log-log models, do not center and scale the response or logged covariates. Instead, interpret regression coefficients as elasticities.*

This rule is particularly useful for economic variables. Often there are published papers that estimate elasticities directly relevant to your problem, and you can use them to help inform your prior.

When you do all these transformations to help you think about the model, you ultimately have to choose priors for the parameters of the transformed model. These priors depend to a large extent on the model, but there are some concrete choices that apply fairly generally. First, before trying to incorporate your background knowledge, you should try to construct default priors.

Rule 8 *Construct default, weakly informative priors first, even if you intend to do the analysis with informative priors.*

It is often useful to see how much prior or auxiliary information is driving your inference. This is not necessarily a bad thing, but it is worth knowing what drives your inferences. Also, recall that from the Bayesian perspective, the prior is part of a model of your uncertainty. In general, it is good practice to compare your favored model against reasonable defaults, and fitting the model with weakly informative priors is one way to do this for your uncertainty model.

It can be attractive to assume that a “flat” prior is a good default. It seems uninformative, because a uniform distribution implies that each area of the parameter space is equally likely, a priori. This turns out to be a bad idea for two main reasons. The smaller issue is that it can often cause computational problems, especially for unbounded parameter spaces and for complicated models. The larger issue is that the intuition that “flat” equals “uninformative” is just wrong. It turns out that *no* prior is globally uninformative. Every prior is informative for some questions. The classic way to see this is transformation. A flat prior for θ implies a prior on θ^2 that puts a lot of mass near zero. A prior can be more or less informative for a particular question, however. The next rule of thumb summarizes this.

Rule 9 *No prior is globally uninformative. Instead, default priors should be weakly informative for the questions that the analyst is trying to answer.*

Without looking at a particular model, you can operationalize default priors to some extent. Generally, they should be centered on the values that you would expect if you had a “default” state of knowledge, and they should be spread out very far. In practice, this depends on the type of parameter that you are considering. Regression coefficients are easiest, as the next rule shows.

Rule 10 *In the absence of other information, a good default prior for a regression coefficient is $\beta_j \sim \mathrm{N}(m, s^2)$, where m is what you expect the estimated coefficient to be, and s is chosen so that you do not expect β_j to be more extreme than $m \pm s$, a priori.*

According to the 68/95/99 rule, this prior says that β_j is more extreme than you thought was possible about 32% of the time. This illustrates just how a weakly informative prior is intended to be—it lightly discourages crazy values of the parameters. This prior assumes that the regression parameter is of direct interest. If you are interested in some function of the regression parameters, use your auxiliary information about that quantity to construct your default priors.

It might be tempting to strictly bound the prior between two extreme values, but in general this is not a good idea. It can cause computational issues, but more important, it is better to allow the model to go into extreme regions of the parameter space if the data are strongly telling it to. For example, you might think the coefficient for the price covariate in a sales regression should be negative, but what if the good is a Giffen good, which consumers buy more of as the price rises? Or if consumers use price as a signal of quality or to signal how wealthy they are? The main exception is for parameters that must be defined in a constrained space by their definition, such as standard deviations and correlations.

Rule 11 *Don't constrain parameters in the prior. Instead, construct priors that regularize away from values that seem impossible but still allow them.*

You can also choose some default values for m and s in Rule 10, though this depends on the model and the type of covariate. The examples cover this detail.

Another common type of parameter is a scale parameter, such as variances, standard deviations, and so on. The conventional wisdom is to use an inverse gamma prior on the variance. However, the inverse gamma prior can be highly informative in ways that are typically undesirable, making it a poor choice for a default prior (see, for example, Gelman 2006). A better choice is a positive truncated normal distribution on the standard deviation, which allows for posterior standard deviations to be arbitrarily close to zero but still penalizes values that seem far too large relative to prior expectations.

Rule 12 *In the absence of other information, use an $\mathrm{N}^+(0, s^2)$ prior on standard deviations, with s set to the upper bound of what you reasonably expect a priori. For other scale parameters, transform to a scale similar to the standard deviation first, then use the $\mathrm{N}^+(0, s^2)$ prior for the transformed parameter.*

By the 68/95/99 rule, this prior implies that the standard deviation is larger than the largest value you could reasonably expect about 32% of the time, which again is only weakly informative about the likely value of that parameter. If in your particular application you do not expect standard deviation values near zero, you can add a mean parameter to the positive truncated normal prior to center it on larger values, such as $\mathrm{N}^+(m, s^2)$, where $m > 0$.

Normal tails are very light, and this affects how the model deals with outliers. When the prior has light tails, the posterior takes outlier observations into account very seriously so that one very large observation can strongly influence the posterior (see, for example, O'Hagan 1979). Typically, it is more desirable to achieve "robust" behavior—that is, when an observation is very extreme relative to the prior and other observations, the posterior places less weight on it. In practice, you can achieve this behavior by using fatter-tailed priors, such as by replacing normals with T distributions.

Rule 13 *To make inferences more robust to outliers, use fatter-tailed priors. For example, replace $\mathrm{N}(m, s^2)$ priors with $\mathrm{T}_d(m, s^2)$ priors and set d to a value somewhere from 3 to 7.*

The degrees of freedom parameter, d, controls how fat the tails are. A larger value implies thinner tails and less robust behavior. Setting d to too small a value can cause computational problems and sometimes break the assumptions that are required for doing Markov chain Monte Carlo simulation (see, for example, Ghosh, Li, and Mitra 2018). So in the absence of a compelling reason to do something different, you should generally set d to at least 3.

EXAMPLE 1: LOG SALES REGRESSION WITH DEFAULT PRIORS

To illustrate how to use the rules of thumb in the previous section, consider a hypothetical network of 100 car dealerships. This network is considering expanding to one of several new locations, and it wants to forecast the yearly sales of a particular model of four-wheel-drive pickup truck—in particular, what impact the price of the truck will have on those sales. As a baseline, the network wants to fit a regression model that uses last year's sales and price data from the existing dealerships in the network to predict the number of trucks that it will sell, controlling for climate and demographic variables for the region. The available variables for each dealership are the type of region it is (**area_type**: rural, suburban, or urban); the number of people in the region who are at least 18 years old and have at least a bachelor's degree (**pop_bachelors**); the number of people in the region who are at least 18 years old and have less than a bachelor's degree (**pop_below_bachelors**); median household income in the region in dollars (**median_income**); cost of living for the region, as measured by an available index (**cost_of_living**, a positive number); average high temperature in degrees Fahrenheit in the summer months—June, July, and August (**mean_summer_temp**); average high temperature in degrees Fahrenheit in the winter months—December, January, and February (**mean_winter_temp**); average yearly precipitation in inches (**mean_precip**); the number of trucks sold (**sales**); and the posted price in dollars (**price**). The following code generates the hypothetical data set:

```
data trucksales;
   call streaminit(768234);
   rural_intercept = 10;
   urban_intercept = 8;
   suburban_intercept = 9;
   do i = 1 to 100;
      population = rand('POISSON', 50000);
      prop_bachelors = rand('BETA', 10, 30);
      pop_bachelors = INT(prop_bachelors * population);
      pop_below_bachelors = INT((1 - prop_bachelors) * population);
      median_income = INT(exp(log(40000) + .3*rand('NORMAL', 0, 1)));
      price = ROUND(25000 + 1000 * rand('NORMAL', 0, 1), 100);
      cost_of_living = INT(130 + 20*rand('NORMAL', 0, 1));
      mean_summer_temp = INT(85 + 5*rand('NORMAL', 0, 1));
      mean_winter_temp = INT(35 + 8*rand('NORMAL', 0, 1));
      mean_precip = INT(exp(log(22) + .4*rand('NORMAL', 0, 1)));
      rural_idx = rand('NORMAL', 0, 1);
      if rural_idx < -0.7 then area_type = 'rural';
      if rural_idx >  0.7 then area_type = 'urban';
      if abs(rural_idx) <= 0.7 then area_type = 'sub';
      if area_type = 'rural' then intercept = rural_intercept;
      if area_type = 'urban' then intercept = urban_intercept;
      if area_type = 'sub'   then intercept = suburban_intercept;
      xbeta = intercept - 1 + 0.03 * log(pop_bachelors) +
         0.04 * log(pop_below_bachelors) + 0.04 * log(median_income) +
         - 0.5 * log(price) - 0.02 * log(cost_of_living) +
         - 0.02 * log(mean_summer_temp) + 0.3 * log(mean_winter_temp) +
         0.02 * log(mean_precip);
      sales = CEIL(exp(xbeta + 0.05 * rand('NORMAL', 0, 1)));
      output;
   end;
   keep pop_bachelors pop_below_bachelors median_income price cost_of_living
      mean_summer_temp mean_winter_temp mean_precip area_type sales;
run; quit;
```

Technically, there is an endogeneity problem here because it is not clear whether the price differences are from different demand curves, different supply curves, or both. For the purposes of this and the later examples, ignore this issue. For example, suppose that the supply curve is perfectly elastic and constant across dealerships, since they are all part of the same network. Keep in mind that these assumptions are very strong and that a more sophisticated model would be required if they do not hold. The following code generates a summary of the data set, which is shown in Figure 1:

```
proc summary data = trucksales print maxdec=2;
   var pop_bachelors pop_below_bachelors median_income cost_of_living
      mean_summer_temp mean_winter_temp mean_precip price sales;
run; quit;

proc summary data = trucksales print;
   class area_type;
run; quit;
```

Figure 1 Summary of Climate, Demographic, and Dealership Data

The SUMMARY Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
pop_bachelors	100	12118.38	2956.35	6684.00	20223.00
pop_below_bachelors	100	37857.67	2969.18	29703.00	43258.00
median_income	100	44012.19	13115.21	18261.00	80122.00
cost_of_living	100	127.36	20.26	78.00	176.00
mean_summer_temp	100	84.70	5.16	71.00	95.00
mean_winter_temp	100	34.19	8.18	11.00	60.00
mean_precip	100	23.83	12.21	5.00	92.00
price	100	25020.00	952.72	22600.00	27500.00
sales	100	177.60	123.79	48.00	469.00

The SUMMARY Procedure

area_type	N Obs
rural	22
sub	52
urban	26

When you fit the regression model, you have several choices to make. Many of the variables are positive constrained—the population variables, income, cost of living, precipitation, price, and sales. It is reasonable to log-transform these variables, especially to take advantage of thinking about them as elasticities in order to construct priors (that is, Rule 7). The temperature variables are not positive constrained, so instead Rule 4 suggests that you should center and scale them. The next bit of code performs all these transformations, then generates a summary of the transformed data set shown in Figure 2:

```
data trucksales_log;
   set trucksales;
   log_pop_bachelors = log(pop_bachelors);
   log_pop_below_bachelors = log(pop_below_bachelors);
   log_median_income = log(median_income);
   log_price = log(price);
   log_cost_of_living = log(cost_of_living);
   log_mean_precip = log(mean_precip);
   log_sales = log(sales);
   mean_summer_temp_cs = mean_summer_temp;
   mean_winter_temp_cs = mean_winter_temp;
   keep mean_summer_temp_cs mean_winter_temp_cs area_type
      log_pop_bachelors log_pop_below_bachelors log_median_income log_price
      log_cost_of_living log_mean_precip log_sales;
run; quit;

proc standard data = trucksales_log mean=0 std=1 out=trucksales_transformed;
   var mean_summer_temp_cs mean_winter_temp_cs;
run; quit;
```

```
proc summary data = trucksales_transformed print maxdec=2;
   var log_pop_bachelors log_pop_below_bachelors log_median_income log_cost_of_living
      mean_summer_temp_cs mean_winter_temp_cs log_mean_precip log_price log_sales;
run; quit;
```

Figure 2 Summary of Transformed Climate, Demographic, and Dealership Data

The SUMMARY Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
log_pop_bachelors	100	9.37	0.25	8.81	9.91
log_pop_below_bachelors	100	10.54	0.08	10.30	10.67
log_median_income	100	10.65	0.30	9.81	11.29
log_cost_of_living	100	4.83	0.16	4.36	5.17
mean_summer_temp_cs	100	-0.00	1.00	-2.66	2.00
mean_winter_temp_cs	100	0.00	1.00	-2.83	3.15
log_mean_precip	100	3.08	0.43	1.61	4.52
log_price	100	10.13	0.04	10.03	10.22
log_sales	100	4.95	0.69	3.87	6.15

Next, you can fit the regression model by using classical methods just to get a baseline. The rationale for doing this is the same as the rationale for Rule 8: if there is a major difference in the analyses, that is worth knowing even if you do not think that it is necessarily a problem. The following code obtains the classical estimates by using PROC QLIM; they are reported in Figure 3.

```
proc qlim data = trucksales_transformed plots = none;
   class area_type;
   model log_sales = area_type log_pop_bachelors log_pop_below_bachelors
                     log_median_income log_price log_cost_of_living
                     log_mean_precip mean_summer_temp_cs mean_winter_temp_cs;
run; quit;
```

Figure 3 Classical Fit to Transformed Truck Sales Data

The QLIM Procedure

Parameter Estimates

Parameter		DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept		1	8.876976	4.165329	2.13	0.0331
area_type	rural	1	2.006472	0.014212	141.18	<.0001
area_type	sub	1	0.998033	0.011482	86.92	<.0001
area_type	urban	0	0	.	.	.
log_pop_bachelors		1	0.094135	0.097595	0.96	0.3348
log_pop_below_bachelors		1	0.112005	0.296659	0.38	0.7058
log_median_income		1	0.016668	0.016753	0.99	0.3198
log_price		1	-0.676310	0.125230	-5.40	<.0001
log_cost_of_living		1	-0.070649	0.029772	-2.37	0.0176
log_mean_precip		1	0.019825	0.011464	1.73	0.0837
mean_summer_temp_cs		1	0.000821	0.004924	0.17	0.8676
mean_winter_temp_cs		1	0.078446	0.004981	15.75	<.0001
_Sigma		1	0.046566	0.003311	14.06	<.0001

Now you can move on to constructing default priors. At this stage, the priors that you select should be fairly generic and independent of what the variable means, unless you have a very strong reason to make a different choice. So use Rule 10 for regression coefficients, and set $m = 0$. The only reasonable exception in this example is the price coefficient, because a positive price elasticity of demand would be very surprising. However, as Rule 8 says, it is a good idea to fit the model with the “standard” default prior anyway, instead of a prior that incorporates this extra information about the coefficient.

For all the logged covariates, you can use Rule 7 and construct a generic prior for all elasticities. Because an elasticity can be positive or negative, m = 0. For s, first consider what would be a very surprising large (or small) value for an elasticity. In many cases, estimated demand elasticities are between –1 and 1—for example, a 1% change in the covariate typically results in less than a 1% change in the amount of the product sold—though sometimes elasticities are more extreme than that, depending on the product and covariate. An elasticity of ± 4 would be very surprising in most contexts, so call that the most extreme value you expect. This prior is weakly informative, since it expects to see an elasticity larger in magnitude than the most extreme value you expect about 32% of the time.

Next, consider the centered and scaled covariates. The easiest way to construct a default prior for these coefficients is to think in terms of standard deviations of the response. This gives you a general, unit-free perspective to work with. So if, for example, the mean summer high temperature increased by one standard deviation, how many standard deviations do you expect log sales to change by? Again, setting m = 0 makes sense—it could increase or decrease sales, but by default you do not know. A ± 4 standard deviation change in log sales would be very surprising, so again you can use that as your choice for s. From Figure 2, you can see that the standard deviation is about 0.69, so set s = $4 \times 0.69 \approx 2.76$.

The next parameters to consider are the dummy variables that are associated with the **area_type** variable. For a default prior, you should have no reason to distinguish among the groups, so each coefficient should be centered on m = 0. To choose s, a good default choice is to suppose that group membership has about the same impact on the response as a change of one standard deviation in a continuous covariate. This implies that s = 2.76.

Finally, you need to choose a prior for the intercept, which under the default dummy encoding in the QLIM procedure corresponds to the intercept for the urban area type. This is trickier, because the value of the intercept varies widely depending on the values of the slopes. So Rule 1 requires you to apply Rules 5 and 6. If you standardized *all* the covariates in the model, then the intercept would be directly interpretable as the unconditional mean of the response variable. In that case, a reasonable default prior would set m equal to the sample mean of the response—that is, 4.95. Typically, you have a combination of dummy variables, standardized continuous covariates, and nonstandardized continuous covariates, as in this example. This makes interpreting the intercept difficult, so it is usually better to set m to the classical estimate of the intercept—that is, 8.88.

To take into account the additional prior uncertainty in the intercept, the value of s should generally be much larger than the values that you choose for the slope coefficients. The largest value of s for any of the regression coefficients was s = 4, so for the intercept s = 100 adds about two orders of magnitude more prior variation. This prior for the intercept builds in an assumption that you are not directly interested in it—that is, it is a nuisance parameter. If you are directly interested in the intercept parameter, then instead of using the classical estimate for m, you should spend more time thinking clearly about what it means for your problem and what you know about it a priori. This approach can be generalized: you should typically put more effort into constructing priors for the parameters that directly matter for your inferential question.

Finally, you need a prior on the error variance—or by Rule 3, the error standard deviation. Rule 12 suggests a good default prior, and you need only choose s. In any regression model, the error standard deviation is almost always less than the response variable's standard deviation, so setting s = 0.69 implies that a priori, you expect the error standard deviation to be above the response's standard deviation about 32% of the time.

All of our priors are listed in one convenient place, as follows. Note that these distributions are assumed to be independent of one another. In principle, you can put dependence in your prior distribution, but in practice it can be quite difficult to think about that dependence and make your intuitions mathematically precise.

$$\begin{aligned}
\beta_{\text{intercept}} &\sim \mathrm{N}(8.88, 100^2) \\
\beta_{\text{log\_pop\_bachelors}}, \beta_{\text{log\_pop\_below\_bachelors}}, \beta_{\text{log\_median\_income}}, & \\
\beta_{\text{log\_cost\_of\_living}}, \beta_{\text{log\_mean\_precip}}, \beta_{\text{log\_price}} &\sim \mathrm{N}(0, 4^2) \\
\beta_{\text{mean\_summer\_temp\_cs}}, \beta_{\text{mean\_winter\_temp\_cs}}, & \\
\beta_{\text{area\_type\_rural}}, \beta_{\text{area\_type\_sub}} &\sim \mathrm{N}(0, 2.76^2) \\
\sigma &\sim \mathrm{N}^{+}(0, 0.69^2)
\end{aligned}$$

The following code fits the model in PROC QLIM by using a joint random walk Metropolis sampler. Figure 4 shows the associated output, including various posterior summaries.

```
proc qlim data = trucksales_transformed plots = none;
   class area_type;
   model log_sales = area_type log_pop_bachelors log_pop_below_bachelors
      log_median_income log_price log_cost_of_living
      log_mean_precip mean_summer_temp_cs mean_winter_temp_cs;
   bayes seed = 72834 ntu = 100 mintune = 20 maxtune = 20 nmc = 10000
      statistics = (summary interval prior);
   prior intercept ~ normal(mean = 8.88, var = 10000);
   prior log_pop_bachelors log_pop_below_bachelors log_median_income
      log_cost_of_living log_mean_precip log_price ~ normal(mean = 0, var = 16);
   prior mean_summer_temp_cs mean_winter_temp_cs
      area_type_rural area_type_sub ~ normal(mean = 0, var = 7.62);
   prior _sigma ~ normal(mean = 0, var = 0.48);
run; quit;
```

Figure 4 Bayesian Fit with Default Priors to Truck Sales Data

The QLIM Procedure

Posterior Summaries

				Percentiles		
Parameter	**N**	**Mean**	**Standard Deviation**	**25%**	**50%**	**75%**
Intercept	10000	9.2469	4.8299	5.9232	9.0512	12.2599
area_type_rural	10000	2.0083	0.0147	1.9979	2.0079	2.0188
area_type_sub	10000	0.9981	0.0127	0.9902	0.9978	1.0062
log_pop_bachelors	10000	0.0865	0.1179	0.00838	0.0975	0.1683
log_pop_below_bachelors	10000	0.0979	0.3387	-0.1114	0.1214	0.3326
log_median_income	10000	0.0194	0.0178	0.00711	0.0194	0.0318
log_price	10000	-0.6945	0.1315	-0.7790	-0.6964	-0.6078
log_cost_of_living	10000	-0.0684	0.0318	-0.0893	-0.0687	-0.0475
log_mean_precip	10000	0.0176	0.0123	0.0105	0.0177	0.0254
mean_summer_temp_cs	10000	0.000351	0.00516	-0.00284	0.000519	0.00384
mean_winter_temp_cs	10000	0.0792	0.00521	0.0754	0.0790	0.0825
_Sigma	10000	0.0495	0.00317	0.0472	0.0494	0.0516

Posterior Intervals

Parameter	**Alpha**	**Equal-Tail Interval**		**HPD Interval**	
Intercept	0.050	0.2670	18.8094	-0.0949	18.3721
area_type_rural	0.050	1.9808	2.0363	1.9808	2.0360
area_type_sub	0.050	0.9720	1.0236	0.9733	1.0239
log_pop_bachelors	0.050	-0.1629	0.2867	-0.1355	0.3065
log_pop_below_bachelors	0.050	-0.6299	0.6927	-0.5930	0.7114
log_median_income	0.050	-0.0130	0.0556	-0.0151	0.0518
log_price	0.050	-0.9639	-0.4380	-0.9490	-0.4269
log_cost_of_living	0.050	-0.1324	-0.00468	-0.1371	-0.0101
log_mean_precip	0.050	-0.00965	0.0415	-0.00652	0.0431
mean_summer_temp_cs	0.050	-0.0106	0.0103	-0.00899	0.0114
mean_winter_temp_cs	0.050	0.0692	0.0904	0.0684	0.0893
_Sigma	0.050	0.0439	0.0560	0.0438	0.0559

Compared to Figure 3, many of the slope estimates in Figure 4 are attenuated toward zero. This small bit of regularization comes from the weakly informative prior and can protect you from making premature inferences, such as from *p*-hacking or the garden of forking paths (for an explanation of the garden, see Gelman and Loken 2014). If this is an explicit goal of the prior, then you should construct it with that purpose in mind. For example, center the priors on the "null hypothesis," and use smaller prior standard deviations to further regularize the parameters.

EXAMPLE 2: PURCHASING DECISION PROBIT REGRESSION WITH DEFAULT PRIORS

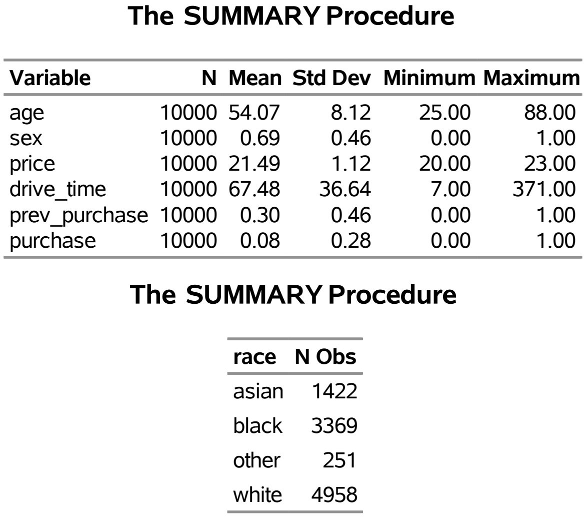
To gather more information, the dealership network wants to fit a probit model by using internal data from an advertising campaign at one dealership. This dealership sent out a flier to 10,000 individuals in its region with an ad for the pickup truck whose sales the network is trying to forecast in Example 1. The fliers were randomized to include one of four possible advertised prices—$20,000, $21,000, $22,000, or $23,000—and were repeatedly sent throughout the year. A recipient would be able to buy the truck at this price only by coming to the dealership with the flier in hand. The dealership recorded whether each individual who received the flier bought a truck at the advertised price over the next year.

Each of the recipients was already in the dealership's advertising database, with several demographic variables recorded. The data set includes the following variables: the race of the recipient (**race**—white, black, Asian, or other), the age of the recipient in years (**age**), whether the recipient is male or female (**sex**—male = 1, female = 0), the amount of time it would take the recipient to drive to the dealership in minutes (**drive_time**), whether the recipient had previously purchased a vehicle at the dealership (**prev_purchase**), the price of the truck in the advertisement in thousands of dollars (**price**), and whether the recipient purchased a truck at the advertised price (**purchase**). The following code generates this data set and summarizes it. The summary is shown in Figure 5.

```
data truck_ad;
   call streaminit(92342);
   do i = 1 to 10000;
      race_idx = rand('NORMAL', 0, 1);
      if race_idx >= 0 then
         do;
            race = 'white';
            age = rand('POISSON', 55);
            intercept = 1.1;
         end;
      if race_idx < 0 then
         do;
            race = 'black';
            age = rand('POISSON', 50);
            intercept = 0.7;
         end;
      if race_idx < -1 then
         do;
            race = 'asian';
            age = rand('POISSON', 60);
            intercept = -2.9;
         end;
      if race_idx < -2 then
         do;
            race = 'other';
            age = rand('POISSON', 55);
            intercept = -1.8;
         end;
      price_idx = rand('uniform', 0, 4);
      price = 20;
      if price_idx > 1 then price = 21;
      if price_idx > 2 then price = 22;
      if price_idx > 3 then price = 23;
      prev_purchase = rand('BINOMIAL', 0.3, 1);
      sex = rand('BINOMIAL', 0.7, 1);
      drive_time = INT(exp(log(60) + 0.5*rand('NORMAL', 0, 1)));
      mu = 7.5 + intercept + 0.1 * sex + 0.001 * age +
         0.002 * prev_purchase - 0.002 * drive_time - 0.5 * price;
      prob = 1 / (1 + exp(-mu));
      purchase = rand('BINOMIAL', prob, 1);
      output;
   end;
   keep race sex age price prev_purchase drive_time purchase;
run; quit;
```

```
proc summary data = truck_ad print maxdec=2;
   var age sex price drive_time prev_purchase purchase;
run; quit;

proc summary data = truck_ad print;
   class race;
run; quit;
```

Figure 5 Summary of Advertisement Data



The SUMMARY Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
age	10000	54.07	8.12	25.00	88.00
sex	10000	0.69	0.46	0.00	1.00
price	10000	21.49	1.12	20.00	23.00
drive_time	10000	67.48	36.64	7.00	371.00
prev_purchase	10000	0.30	0.46	0.00	1.00
purchase	10000	0.08	0.28	0.00	1.00

The SUMMARY Procedure

race	N Obs
asian	1422
black	3369
other	251
white	4958

Your task is to fit a probit model by using these variables to estimate the impact of the advertised price on whether or not a recipient will respond to the ad. This information is then used to inform the prior on the log price variable from Example 1. The probit model assumes that

$$P(y_i = 1) = \Phi(\mathrm{x}_i' \boldsymbol{\beta})$$

for $i = 1, 2, \ldots, 10,000$, where y_i is the recipient's **purchase** variable, x_i is a vector of the recipient's covariates listed earlier, $\boldsymbol{\beta}$ is a corresponding vector of regression coefficients, and $\Phi()$ is the standard normal cumulative distribution function. Your task for this example is to come up with a reasonable default prior for $\boldsymbol{\beta}$.

First, Rules 4 and 7 apply. Although this model is not a log-log model, the results of the model will be used to inform the prior on an elasticity, so it is convenient to make the mean structures of the two models similar. Another reason not to standardize **price** is that the variable's variation in the data set is in some sense artificial: it was chosen deliberately by the designers of the ad campaign and does not necessarily represent real-world variation in price. The other continuous variables—**age** and **drive_time**—are also positive constrained, but in this model they are easier to think about when centered and scaled than when log-transformed. The following code transforms each variable as appropriate and produces the summary of the transformed data set shown in Figure 6:

```
data truck_ad_cs;
   set truck_ad;
   age_cs = age;
   drive_time_cs = drive_time;
   log_price = log(price);
   keep race sex age_cs drive_time_cs prev_purchase log_price purchase;
run; quit;

proc standard data = truck_ad_cs mean=0 std=1 out=truck_ad_transformed;
   var age_cs drive_time_cs;
run; quit;

proc summary data = truck_ad_transformed print maxdec=2;
   var age_cs sex log_price drive_time_cs prev_purchase purchase;
run; quit;
```

Figure 6 Summary of Advertisement Data

The SUMMARY Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
age_cs	10000	-0.00	1.00	-3.58	4.18
sex	10000	0.69	0.46	0.00	1.00
log_price	10000	3.07	0.05	3.00	3.14
drive_time_cs	10000	0.00	1.00	-1.65	8.28
prev_purchase	10000	0.30	0.46	0.00	1.00
purchase	10000	0.08	0.28	0.00	1.00

Next, the following code fits the model by using classical methods. Figure 7 shows the resulting parameter estimates.

```
proc qlim data = truck_ad_transformed plots = none;
   class purchase race;
   model purchase = race sex age_cs drive_time_cs prev_purchase log_price
      / discrete(dist = normal);
run; quit;
```

Figure 7 Probit Fit to Transformed Advertisement Data

The QLIM Procedure

Parameter Estimates

Parameter		DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept		1	14.742479	1.154745	12.77	<.0001
race	asian	1	-2.045956	0.293640	-6.97	<.0001
race	black	1	-0.171212	0.041751	-4.10	<.0001
race	other	1	-1.239475	0.268596	-4.61	<.0001
race	white	0	0	.	.	.
sex		1	0.054499	0.041671	1.31	0.1909
age_cs		1	0.023252	0.020944	1.11	0.2669
drive_time_cs		1	-0.024131	0.019554	-1.23	0.2172
prev_purchase		1	-0.038599	0.042132	-0.92	0.3596
log_price		1	-5.228477	0.378214	-13.82	<.0001

To construct a prior for the continuous covariates, first think about what they mean in the context of the model. Following Rule 6, suppose that for some base case recipient, $\mathbf{x}_i'\boldsymbol{\beta} = \mu_i$. Now suppose that the age of the recipient increases by one standard deviation. Then the change in the probability that the recipient purchases the truck, denoted by Δ, is given by

$$\Delta = \Phi(\mu_i + \beta_{\text{age\_cs}}) - \Phi(\mu_i)$$

Solving for $\beta_{\text{age\_cs}}$ yields

$$\beta_{\text{age\_cs}} = \Phi^{-1}\left[\Delta + \Phi(\mu_i)\right] - \mu_i$$

where $\Phi^{-1}()$ is the standard normal quantile function. You can use this equation to choose m and s for use in Rule 10. A useful base case is a recipient who is about 50/50 on whether to purchase the advertised truck, which implies $\mu_i = 0$. Now increase this person's age by one sample standard deviation. How much do you expect the probability that the person will purchase the truck to change? Call this value Δ_M. Then, plugging everything into the preceding equation yields a value for m in the prior for $\beta_{\text{age\_cs}}$. A good default is $\Delta_M = 0$, which yields

$$m = \Phi^{-1}(\Delta_M + \Phi(0)) = \Phi^{-1}(0.5) = 0$$

To get a value for s, what is the largest Δ you would expect after a change in **age** of one standard deviation? What about the smallest Δ? Call these Δ_U and Δ_L, respectively. Then identify them with a one-standard-deviation increase and decrease relative to m, respectively, to get the following equations:

$$m + s = \Phi^{-1}(\Delta_U + 0.5)$$
$$m - s = \Phi^{-1}(\Delta_L + 0.5)$$

Rearranging and plugging in m = 0 yields

$$s = \Phi^{-1}(\Delta_U + 0.5)$$
$$s = -\Phi^{-1}(\Delta_L + 0.5)$$

A good default here is $\Delta_U = 0.3 = -\Delta_L$. Because the maximum possible change in $P(y_i = 1)$ from the base case is ± 0.5, this would be a very large change relative to such a small change in the covariate. Plugging it into the equation yields

$$s = \Phi^{-1}(0.8) \approx 0.84$$
$$s = -\Phi^{-1}(0.2) \approx 0.84$$

Note that $\Phi^{-1}(0.2) = -\Phi^{-1}(0.8)$, so it does not matter whether you take the most positive or most negative extreme value for Δ, but in general you should take the maximum value for s implied by the two equations. All of this reasoning applies for $\beta_{\text{drive\_time}}$ as well, because this is a default prior stated in terms of the sample standard deviation. Thus the independent priors for both variables are

$$\beta_{\text{age\_cs}}, \beta_{\text{drive\_time\_cs}} \sim \text{N}(0, 0.84^2)$$

The elasticity reasoning for **log_price** is complicated in this model, and as discussed earlier, thinking in terms of the sample standard deviation is also not useful. Instead, consider a base case again with **log_price** = $\log 20$, and suppose that the price increases by exactly $1,000. Then set up the equation for the change in $P(y_i = 1)$:

$$\Delta = \Phi\left[\mu_i + (\log 21 - \log 20)\beta_{\text{log\_price}}\right]) - \Phi(\mu_i)$$
$$\approx \Phi(\mu_i + 0.05\beta_{\text{log\_price}}) - \Phi(\mu_i)$$

Again, plug in $\mu_i = 0$ to make the base case as simple as possible, and solve for $\beta_{\text{log\_price}}$ to obtain

$$\beta_{\text{log\_price}} \approx 20\Phi^{-1}(\Delta + 0.5)$$

Now use the same tricks as before. Set $\Delta_M = 0$ to obtain m = 0. Then set $\Delta_U = 0.3 = -\Delta_L$ to obtain

$$m + s = 20\Phi^{-1}(\Delta_U + 0.5)$$
$$s = 20\Phi^{-1}(0.8) \approx 20 \times 0.84 \approx 16.8$$

This produces a default prior of $\beta_{\text{log\_price}} \sim \text{N}(0, 16.8^2)$.

For the coefficients on dummy covariates, **sex**, **prev_purchase**, and the race dummies, most of the work is already done if you can easily compare them to the coefficients that you already have priors for. How much of an impact do you expect a change in the dummy variable to have on $P(y_i = 1)$, relative to a change of one standard deviation in either **age** or **drive_time**? A good default answer is "about the same." This yields the independent priors

$$\beta_{\text{sex}}, \beta_{\text{prev\_purchase}} \sim \text{N}(0, 0.84^2)$$

Finally, you need a prior for the intercept. As in Example 1, a good default for m when all covariates have been standardized is the sample mean response. In this case, that response is a rate and needs to be transformed by the link function—that is, m = $\Phi^{-1}(0.08) = -1.41$. In this example, where some covariates are standardized and others are not, a better default is the classical estimate for the intercept, so m = 14.74. Again, this default prior works best when the intercept is not directly related to your inferential questions.

To choose s, once again set it much larger than the largest slope coefficient standard deviation, which is s = 16.8. For example s = 100 should provide plenty of prior variation to accommodate a wide range of possibilities. This yields the following full set of priors:

$$\beta_{\text{intercept}} \sim \text{N}(14.74, 100^2)$$
$$\beta_{\text{age\_cs}}, \beta_{\text{drive\_time\_cs}}, \beta_{\text{sex}}, \beta_{\text{prev\_purchase}},$$
$$\beta_{\text{asian}}, \beta_{\text{black}}, \beta_{\text{other}} \sim \text{N}(0, 0.84^2)$$
$$\beta_{\text{log\_price}} \sim \text{N}(0, 16.8^2)$$

The following code fits the model in PROC QLIM by using the default priors listed earlier and the random walk Metropolis sampler, and it produces the summaries of the posterior shown in Figure 8:

```
proc qlim data = truck_ad_transformed plots = none;
   class purchase race;
   model purchase = race sex age_cs drive_time_cs prev_purchase log_price
      / discrete(dist = normal);
   bayes seed = 2341685 ntu = 100 mintune = 20 maxtune = 20 nmc = 10000
      statistics = (summary interval prior);
   prior intercept ~ normal(mean = 14.74, var = 10000);
   prior age_cs drive_time_cs sex prev_purchase
      race_asian race_black race_other ~ normal(mean = 0, var = 0.71);
   prior log_price ~ normal(mean = 0, var = 283);
run; quit;
```

Figure 8 Bayesian Fit with Default Priors to Advertisement Data

The QLIM Procedure

Posterior Summaries

				Percentiles		
Parameter	**N**	**Mean**	**Standard Deviation**	**25%**	**50%**	**75%**
Intercept	10000	14.7989	1.0583	14.0680	14.8058	15.5272
race_asian	10000	-1.9033	0.2295	-2.0414	-1.8778	-1.7452
race_black	10000	-0.1728	0.0430	-0.2019	-0.1712	-0.1433
race_other	10000	-1.1579	0.2520	-1.3205	-1.1419	-0.9766
sex	10000	0.0569	0.0421	0.0312	0.0567	0.0862
age_cs	10000	0.0224	0.0214	0.00834	0.0236	0.0369
drive_time_cs	10000	-0.0226	0.0192	-0.0359	-0.0229	-0.00982
prev_purchase	10000	-0.0399	0.0399	-0.0681	-0.0411	-0.0132
log_price	10000	-5.2482	0.3453	-5.4846	-5.2482	-5.0090

Posterior Intervals

Parameter	**Alpha**	**Equal-Tail Interval**		**HPD Interval**	
Intercept	0.050	12.7640	16.9574	12.7596	16.9074
race_asian	0.050	-2.4183	-1.5028	-2.3601	-1.4650
race_black	0.050	-0.2614	-0.0894	-0.2627	-0.0924
race_other	0.050	-1.7089	-0.7257	-1.6214	-0.6623
sex	0.050	-0.0302	0.1375	-0.0285	0.1379
age_cs	0.050	-0.0190	0.0651	-0.0177	0.0653
drive_time_cs	0.050	-0.0593	0.0159	-0.0580	0.0165
prev_purchase	0.050	-0.1188	0.0424	-0.1097	0.0446
log_price	0.050	-5.9548	-4.5839	-5.9167	-4.5661

As in Example 1, this posterior attenuates some of the slope estimates toward zero, though not all of them. But it still serves as a useful baseline for building more informative priors, if that is your goal with this analysis.

EXAMPLE 3: SALES COUNT REGRESSION WITH AN INFORMATIVE PRICE PRIOR

Next, the dealership network wants to improve the original regression model from Example 1 in two ways. First, management wants you to use a count regression model that takes into account the fact that sales is an integer; and second, it wants you to use an informative prior for price that is informed by the probit fit from Example 2.

The basic count regression model is a Poisson regression where

$$y_i \overset{\text{ind}}{\sim} \text{Poisson}(\lambda_i)$$

and $\log \lambda_i = \mathbf{x}_i' \boldsymbol{\beta}$. This model's response variable is **sales** instead of **log_sales**, but the log-link function that defines λ_i enables you to continue to interpret the coefficients for logged covariates as elasticities, at least approximately. This will come in handy when it is time to construct priors. In fact, apart from the coefficient for **log_price**, you can just use the default priors from Example 1 here. This model has no standard deviation or other dispersion parameter, so you can ignore the original regression model's prior for σ.

In an ideal world, a Bayesian analyst might try to build a joint model of the two data sets so that information learned from one data set can spill over into the fit to the other data set, and vice versa. But this is essentially impossible. The two models control for different sets of variables and have different types of response variables, and the advertisement data came from only one of the dealerships represented in the sales data set. Generalizing to every other dealership is nearly impossible, and converting from the context of one set of covariates to another set is even harder—not to mention that such a model can introduce computational difficulties.

Constructing an informative prior for $\beta_{\text{log\_price}}$ from the results of the probit fit is still not straightforward, but not requiring a precise statement of the connection between the two data sets makes it easier. The probit fit tells you *something* about the price elasticity of demand for the truck model, though it is not clear precisely what. You can compensate for this uncertainty by making the prior distribution relatively less certain about the value of the relevant parameter.

The first challenge is to convert information about $\beta_{\text{log\_price}}$ from the probit model to information about an *elasticity*. Suppose that there are N individuals in a given region, and suppose that according to the probit model the ith individual's probability of purchasing the truck is $\Phi(\mu_i)$. Then the expected number of purchases (that is, the expected demand) is $E = \sum_{i=1}^{N} \Phi(\mu_i)$.

To get an elasticity with respect to price, you need the derivative of $\log E$ with respect to **log_price**,

$$\frac{\partial \log E}{\partial \text{log\_price}} = \frac{\sum_{i=1}^{N} \phi(\mu_i)}{\sum_{i=1}^{N} \Phi(\mu_i)} \beta_{\text{log\_price}}$$

where $\phi()$ is the standard normal probability density function.

A simple way to get a value for this elasticity is to set each $\Phi(\mu_i)$ to the sample mean **purchase** rate, which implies $\mu_i = \Phi^{-1}(0.084) \approx -1.38$ for all i. It would be more accurate to plug in the values of the covariates for each member of the population, but setting everyone at the sample mean makes the calculation easier and means that it does not depend on the population size. Plugging $\mu_i = -1.38$ and the posterior mean of $\beta_{\text{log\_price}}$ into the elasticity equation yields

$$\frac{\partial \log E}{\partial \text{log\_price}} \approx \frac{\phi(-1.38)}{\Phi(-1.38)} \beta_{\text{log\_price}} \approx 1.84 \times -5.22 \approx -9.60$$

This is an overestimate, since the estimate for $\beta_{\text{log\_price}}$ is coming from a model that has data only from people who you are pretty sure knew about the price. Many people in the region do not know about a dealership's price changes one way or another, in which case their price elasticity of demand is zero. Suppose only one in ten individuals in the region learn any information about the price, whether through a flier in the mail, word of mouth, or some other means. Then the price elasticity of demand is about –0.96. According to the literature, this seems reasonable. For example, Copeland (2009) finds a price elasticity of demand for GMC pickup trucks of about –1 in a dynamic model and –2 in a simpler static model. So using Rule 10, you can set m = –0.96.

To choose s, it is instructive to start with the original weakly informative prior from the regression model, in that case with s = 4. In this case, you have stronger prior information about the likely value of the elasticity, so it is reasonable to tighten down the prior. Setting s = 0.5 provides an informative prior, but not too informative. It assumes that there is about a 95% chance that the elasticity is between 0 and –2. A positive elasticity would be very surprising, and although an elasticity of about –2 would be in line with the estimates from the dynamic model dynamic models in Copeland (2009), it is very small relative to the implied elasticity from the probit fit. With these considerations in mind, a reasonable prior for $\beta_{\text{log\_price}}$ is then

$$\beta_{\text{log\_price}} \sim \mathrm{N}(-0.96, 0.5^2)$$

The following code preprocesses the data and fits the model by using the COUNTREG procedure. The resulting classical estimates are shown in Figure 9, and the posterior summaries are shown in Figure 10.

```
data trucksales_count;
   set trucksales;
   log_pop_bachelors = log(pop_bachelors);
   log_pop_below_bachelors = log(pop_below_bachelors);
   log_median_income = log(median_income);
   log_price = log(price);
   log_cost_of_living = log(cost_of_living);
   log_mean_precip = log(mean_precip);
   mean_summer_temp_cs = mean_summer_temp;
   mean_winter_temp_cs = mean_winter_temp;
   keep mean_summer_temp_cs mean_winter_temp_cs area_type
      log_pop_bachelors log_pop_below_bachelors log_median_income log_price
      log_cost_of_living log_mean_precip sales;
run; quit;

proc standard data = trucksales_count mean=0 std=1 out=truckcount_transformed;
   var mean_summer_temp_cs mean_winter_temp_cs;
run; quit;

proc countreg data = truckcount_transformed plots = none;
   class area_type;
   model sales = area_type log_pop_bachelors log_pop_below_bachelors
      log_median_income log_price log_cost_of_living
      log_mean_precip mean_summer_temp_cs mean_winter_temp_cs;
   bayes seed = 56549 ntu = 100 mintune = 20 maxtune = 20 nmc = 10000
      statistics = (summary interval prior);
   prior intercept ~ normal(mean = 8.88, var = 10000);
   prior log_pop_bachelors log_pop_below_bachelors log_median_income
      log_cost_of_living log_mean_precip log_price ~ normal(mean = 0, var = 16);
   prior mean_summer_temp_cs mean_winter_temp_cs
      area_type_rural area_type_sub ~ normal(mean = 0, var = 7.62);
   prior log_price ~ normal(mean = -0.96, var = 0.25);
run; quit;
```

Figure 9 Classical Poisson Regression Estimates of Truck Sales Data

The COUNTREG Procedure

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Approx Pr > |t|
Intercept	1	10.560301	6.442817	1.64	0.1012
area_type rural	1	2.001451	0.029271	68.38	<.0001
area_type sub	1	0.999953	0.028986	34.50	<.0001
area_type urban	0	0	.	.	.
log_pop_bachelors	1	0.035539	0.149601	0.24	0.8122
log_pop_below_bachelors	1	-0.064428	0.454199	-0.14	0.8872
log_median_income	1	0.009019	0.028098	0.32	0.7482
log_price	1	-0.598401	0.195036	-3.07	0.0022
log_cost_of_living	1	-0.061695	0.050142	-1.23	0.2185
log_mean_precip	1	0.011931	0.020699	0.58	0.5644
mean_summer_temp_cs	1	-0.006239	0.008432	-0.74	0.4593
mean_winter_temp_cs	1	0.079396	0.008708	9.12	<.0001

Figure 10 Bayesian Poisson Regression Estimates of Truck Sales Data

Posterior Summaries

				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
Intercept	10000	10.7170	6.2750	6.4443	10.7617	14.8108
area_type_rural	10000	2.0039	0.0308	1.9821	2.0043	2.0242
area_type_sub	10000	1.0032	0.0311	0.9824	1.0019	1.0238
log_pop_bachelors	10000	0.0431	0.1475	-0.0540	0.0435	0.1437
log_pop_below_bachelors	10000	-0.0178	0.4491	-0.3012	-0.0143	0.2869
log_median_income	10000	0.0111	0.0278	-0.00814	0.0103	0.0298
log_price	10000	-0.6738	0.1832	-0.8056	-0.6749	-0.5578
log_cost_of_living	10000	-0.0584	0.0495	-0.0916	-0.0586	-0.0235
log_mean_precip	10000	0.0128	0.0200	-0.00072	0.0130	0.0270
mean_summer_temp_cs	10000	-0.00622	0.00848	-0.0122	-0.00598	-0.00022
mean_winter_temp_cs	10000	0.0792	0.00880	0.0733	0.0796	0.0854

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	-1.5525	23.3088	-1.6171	22.7344
area_type_rural	0.050	1.9439	2.0657	1.9424	2.0634
area_type_sub	0.050	0.9404	1.0634	0.9385	1.0590
log_pop_bachelors	0.050	-0.2460	0.3203	-0.2460	0.3203
log_pop_below_bachelors	0.050	-0.9247	0.8337	-0.9476	0.8073
log_median_income	0.050	-0.0435	0.0659	-0.0414	0.0669
log_price	0.050	-1.0203	-0.3003	-1.0536	-0.3428
log_cost_of_living	0.050	-0.1542	0.0358	-0.1502	0.0365
log_mean_precip	0.050	-0.0269	0.0522	-0.0233	0.0545
mean_summer_temp_cs	0.050	-0.0226	0.0104	-0.0232	0.00946
mean_winter_temp_cs	0.050	0.0612	0.0966	0.0624	0.0972

In this case, the classical estimate of the price elasticity of demand for the truck is around –0.6, whereas the Bayesian estimate with the informative prior is around –0.67. The prior attenuates the estimate toward –1 somewhat, but not a large amount. Because the literature suggests that price elasticities of demand for cars and trucks are typically –1 or –2, and the advertising data suggest that it is about –1, this should improve the quality of your inferences.

The posterior standard deviation of $\beta_{\text{log\_price}}$ is also a bit smaller than the parameter's standard error in the classical estimation. As a result, Bayesian credible intervals are somewhat narrower than classical confidence intervals. From Figure 10, you see that the 95% credible interval for $\beta_{\text{log\_price}}$ is about $(-1.02, -0.30)$, whereas the 95% confidence interval can be computed from Figure 9 as $(-0.99, -0.21)$. If you trust the information in your prior, this should be regarded as a feature of the Bayesian analysis. The credible interval is narrower, to reflect the fact that the prior and data largely agree on the likely values of the parameter, though they do not completely agree.

DISCUSSION

In each of the three examples, several choices need to be made in order to come up with a prior distribution, and typically there is no one right choice. This is an unfortunate reality of statistics: your choices in the data selection, data preprocessing, model selection, and prior selection steps can have a major impact on your results, but in many cases you have only rough guidelines at best. Prior selection is a unique issue in the Bayesian context, but non-Bayesians have their own set of similar issues, such as choosing significance levels for hypothesis tests or minimum detectable effects in power analyses.

There is typically no one best way to make these choices, and indeed there is no one best way to construct priors. No theorem can tell you how to convert vague intuitions that are based on your experience into mathematically precise probability distributions. For other examples of how to navigate these waters, see, for example, Kadane and Wolfson (1998), O'Hagan (1998), or Albert et al. (2012). An alternative approach is to construct "reference" or "objective" priors that satisfy some desirable properties in the case where you have no useful information with which to inform your prior distributions. Two common approaches here are choosing priors that satisfy some intuitive mathematical definition of "ignorance," and choosing priors that result in posterior credible intervals that match the classical confidence intervals. See, for example, Berger and Bernardo (1992) and Berger (2006) for more information about these approaches.

The approach in this paper is to use mathematical tricks, usually transformations, to make quantities in the model easier to think about. Then the analyst can attempt to translate outside knowledge about the problem into a prior distribution on the parameters. By its nature, this process must be ad hoc. This paper presents a number of rules of thumb to guide you through the process, but they should not be taken as general rules that must always apply. The examples illustrate how to apply some of these rules, but the particular choices made there should not be taken as canonical for those examples. There is always room for disagreement on how to best represent prior information, and the goals that you have for fitting the model will further inform those priors. It is good practice to fit the model with several priors that seem to reasonably represent your prior information, whether that information comes from intuitions about the problem or vaguely related data sets. Comparing the results shows you just how sensitive your inferences are to how you translated vague prior information into prior distributions.

Several benefits of taking the Bayesian approach also emerge. First, Bayesian inference enables the analyst to make probability statements about quantities of interest. A 95% credible interval for a parameter is an interval such that the probability that the parameter is in the interval is 0.95, at least according to the model. This is typically much easier for the consumers of model output to interpret than confidence intervals. Second, even weakly informative priors tend to attenuate parameter estimates toward their default values. When set correctly, these priors help protect you from, for example, multiple comparisons issues. Third, when the model and the prior agree on the likely values of a parameter, the resulting posterior distribution is more concentrated around those likely values. If the prior is set well, this is faithfully showing you that your uncertainty about the parameter should decrease. Finally, the process of designing priors that accurately represent your uncertainty, or even a default state of uncertainty, forces you to interrogate your beliefs about the problem and the model. This can lead to new insights into how to think about the problem, potentially resulting in better models, and can often highlight the weak points of the analysis, including the priors. Often this is the greatest benefit of a Bayesian approach: it forces you to think more clearly and be more transparent about the assumptions that go into your analysis.

REFERENCES

Albert, I., Donnet, S., Guihenneuc-Jouyaux, C., Low Choy, S., Mengersen, K. L., and Rousseau, J. (2012). "Combining Expert Opinions in Prior Elicitation." *Bayesian Analysis* 7:503–532.

Berger, J. O. (2006). "The Case for Objective Bayesian Analysis." *Bayesian Analysis* 3:385–402. http://www.stat.cmu.edu/bayesworkshop/2005/berger.pdf.

Berger, J. O., and Bernardo, J. M. (1992). "On the Development of the Reference Prior Method." In *Bayesian Statistics 4: Proceedings of the Fourth Valencia International Meeting*, edited by J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, 35–60. Oxford: Oxford University Press.

Copeland, A. M. (2009). *The Dynamics of Automobile Expenditures*. Federal Reserve Bank of New York Staff Report 394.

Gelman, A. (2006). "Prior Distributions for Variance Parameters in Hierarchical Models." *Bayesian Analysis* 1:515–533.

Gelman, A., and Loken, E. (2014). "The Statistical Crisis in Science." *American Scientist* 102:460–466.

Ghosh, J., Li, Y., and Mitra, R. (2018). "On the Use of Cauchy Prior Distributions for Bayesian Logistic Regression." *Bayesian Analysis* 13:359–383.

Kadane, J. B., and Wolfson, L. J. (1998). "Experiences in Elicitation." *Journal of the Royal Statistical Society, Series D* 47:3–19.

O'Hagan, A. (1979). "On Outlier Rejection Phenomena in Bayes Inference." *Journal of the Royal Statistical Society, Series B* 41:358–367.

O'Hagan, A. (1998). "Eliciting Expert Beliefs in Substantial Practical Applications." *Journal of the Royal Statistical Society, Series D* 47:21–35.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Matthew Simpson
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
Matt.Simpson@sas.com