Practical Bayesian Computation using SAS®

Fang Chen
SAS Institute Inc.
fangk.chen@sas.com

ASA Conference on Statistical Practices
February 20, 2014

Learning Objectives

Attendees will

- understand basic concepts and computational methods of Bayesian statistics
- be able to deal with some practical issues that arise from Bayesian analysis
- be able to program using SAS/STAT procedures with Bayesian capabilities to implement various Bayesian models.

1. Introduction to Bayesian statistics
   - Background and concepts in Bayesian methods
   - Prior distributions
   - Computational Methods
     - Gibbs Sampler
     - Metropolis Algorithm
   - Practical Issues in MCMC
     - Convergence Diagnostics

2. The GENMOD, PHREG, LIFEREG, and FMM Procedures
   - Overview of Bayesian capabilities in the GENMOD, PHREG, LIFEREG, and FMM procedures
   - Prior distributions
   - The BAYES statement
   - GENMOD: linear regression
   - GENMOD: binomial model
   - PHREG: Cox model
   - PHREG: piecewise exponential model (optional)

3. The MCMC Procedure
   - A Primer on PROC MCMC
   - Monte Carlo Simulation
   - Single-level Model: Hyperparameters
   - Generalized Linear Models
   - Random-effects models
     - Introduction
     - Logistic Regression - Overdispersion
     - Hyperpriors in Random-Effects Models - Shrinkage
     - Repeated Measurements Models
   - Missing Data Analysis
     - Introduction
     - Bivariate Normal with Partial Missing
     - Nonignorable Missing (Selection Model)
   - Survival Analysis (Optional)
     - Piecewise Exponential Model with Frailty

Statistics and Bayesian Statistics

- What is Statistics:
  - the science of learning from data, which includes the aspects of collecting, analyzing, interpreting, and *communicating uncertainty*.
- What is Bayesian Statistics:
  - a subset of statistics in which *all uncertainties* are summarized through *probability distributions*.

The Bayesian Method

Given data $\mathbf{x}$, Bayesian inference is carried out in the following way:

1. You select a model (likelihood function) $f(\mathbf{x}|\theta)$ to describe the distribution of $\mathbf{x}$ given θ.
2. You choose a prior distribution $\pi(\theta)$ for θ.
3. You update your beliefs about θ by combining information from $\pi(\theta)$ and $f(\mathbf{x}|\theta)$ and obtain the posterior distribution $\pi(\theta|\mathbf{x})$.

The paradigm can be thought as a transformation from the **before** to the **after**:

$$\pi(\theta) \longrightarrow \pi(\theta|\mathbf{x})$$

Bayes' Theorem

The updating of beliefs is carried out by using Bayes' theorem:

$$\pi(\theta|\mathbf{x}) = \frac{\pi(\theta, \mathbf{x})}{\pi(\mathbf{x})} = \frac{f(\mathbf{x}|\theta)\pi(\theta)}{\pi(\mathbf{x})} = \frac{f(\mathbf{x}|\theta)\pi(\theta)}{\int f(\mathbf{x}|\theta)\pi(\theta)d\theta}$$

The marginal distribution $\pi(\mathbf{x})$ is an integral that is often ignored (as long as it is finite). Hence $\pi(\theta|\mathbf{x})$ is often written as:

$$\pi(\theta|\mathbf{x}) \propto f(\mathbf{x}|\theta)\pi(\theta) = \mathcal{L}(\theta)\pi(\theta)$$

All inferences are based on the posterior distribution.

Two Different Paradigms[1]

Bayesian

- Probability describes degree of belief, not limiting frequency. It is subjective.
- Parameters cannot be determined exactly. They are random variables, and you can make probability statements about them.
- Inferences about θ are based on the probability distribution for the parameter.

Frequentist/Classical

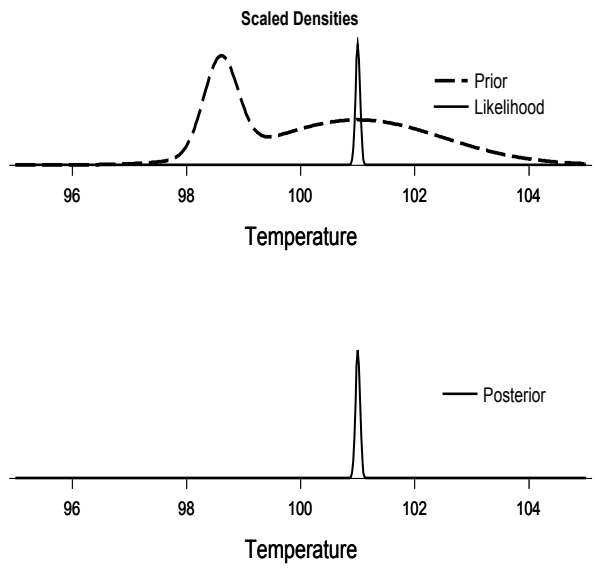
- Probabilities are objective properties of the real world. Probability refers to limiting relative frequencies.
- Parameters θ are fixed, unknown constants.
- Statistical procedures should be designed to have well-defined long-run frequency properties, such as the confidence interval.

[1] Wasserman 2004

Bayesian Thinking in Real Life

You suspect you might have a fever and decide to take your temperature.

1. A possible prior density on your temperature θ: likely normal (centered at 98.6) but possibly sick (centered at 101).
2. Suppose the thermometer says 101 degrees: $f(x|\theta) \sim N(\theta, \sigma^2)$ where σ could be a very small number.
3. You get the posterior distribution. Yes, you are sick.



Estimations

All inference about θ is based on $\pi(\theta|\mathbf{x})$.

- Point: mean, mode, median, any point from $\pi(\theta|\mathbf{x})$. For example, the posterior mean of θ is $E(\theta|\mathbf{x}) = \int_\Theta \theta \cdot \pi(\theta|\mathbf{x}) d\theta$
  The posterior mode of θ is the value of θ that maximizes $\pi(\theta|\mathbf{x})$.
- Interval: credible sets are any set A such that
  $P(\theta \in A|\mathbf{x}) = \int_A \pi(\theta|\mathbf{x}) d\theta$
  - Equal tail: $100(\alpha/2)$th and $100(1-\alpha/2)$th percentiles.
  - Highest posterior density (HPD):
    1. Posterior probability is $100(1-\alpha)\%$
    2. For $\theta_1 \in A$ and $\theta_2 \notin A$, $\pi(\theta_1|\mathbf{x}) \geq \pi(\theta_2|\mathbf{x})$. The smallest region can be disjoint.

  Interpretation: "There is a 95% chance that the parameter is in this interval." The parameter is random, not fixed.

Prior Distributions

The prior distribution represents your belief *before* seeing the data.

- Bayesian probability measures the degree of belief that you have in a random event. By this definition, probability is highly subjective. It follows that all priors are *subjective* priors.
- Not everyone agrees with the preceding. Some people would like to obtain results that are objectively valid, such as, "Let the data speak for itself.". This approach advocates noninformative (flat/improper/Jeffreys) priors.
- Subjective approach advocates informative priors, which can be extraordinarily useful, if used correctly.
- Generally speaking, as the amount of data grows (in a model with fixed number of parameters), the likelihood overwhelms the impact of the prior.

Noninformative Priors

- A prior is *noninformative* if it is *flat* relative to the likelihood function. Thus, a prior $\pi(\theta)$ is noninformative if it has minimal impact on the posterior of θ.
- Many people like noninformative priors because they appear to be more objective. However, it is unrealistic to think that noninformative priors represent total ignorance about the parameter of interest. See Kass and Wasserman (1996): JASA: 91:1343-1370.
- A frequent noninformative prior is $\pi(\theta) \propto 1$, which assigns equal likelihood to all possible values of the parameter.
  - However, flat prior is not invariant: flat on odds ratio is not the same as flat on log of odds ratio.

A Binomial Example

- Suppose that you observe 14 heads in 17 tosses. The likelihood is:

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

with $x = 14$ and $n = 17$.

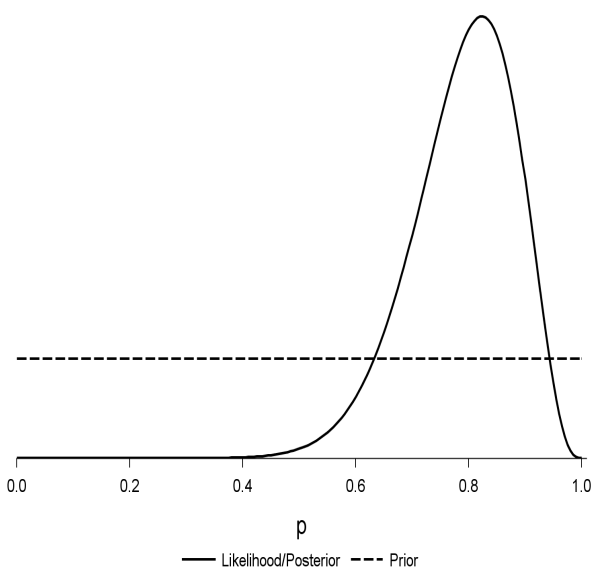
- A flat prior on p is:

$$\pi(p) = 1$$

- The posterior distribution is:

$$\pi(p|x) \propto p^{14}(1-p)^3$$

which is a beta(15, 4).

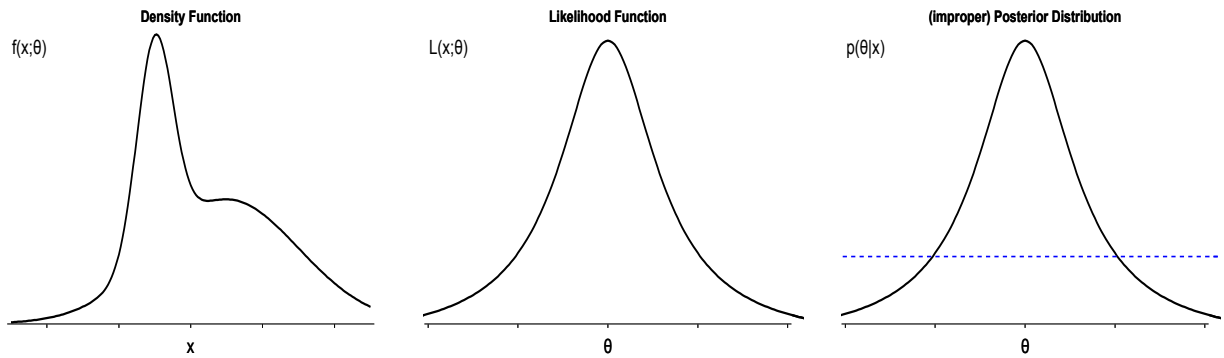


Flat Prior (Observation I)

If $\pi(\theta|\mathbf{x}) \propto \mathcal{L}(\theta)$ with $\pi(\theta) \propto 1$, then why not use the flat prior all the time?

- Using a flat prior does not always guarantee a proper (integrable) posterior distribution; that is, $\int \pi(\theta|\mathbf{x})d\theta < \infty$.

The reason is that the likelihood function is only proper w.r.t. the random variable $\mathbf{X}$. But a posterior has to be integrable w.r.t. θ, a condition not required by the likelihood function.

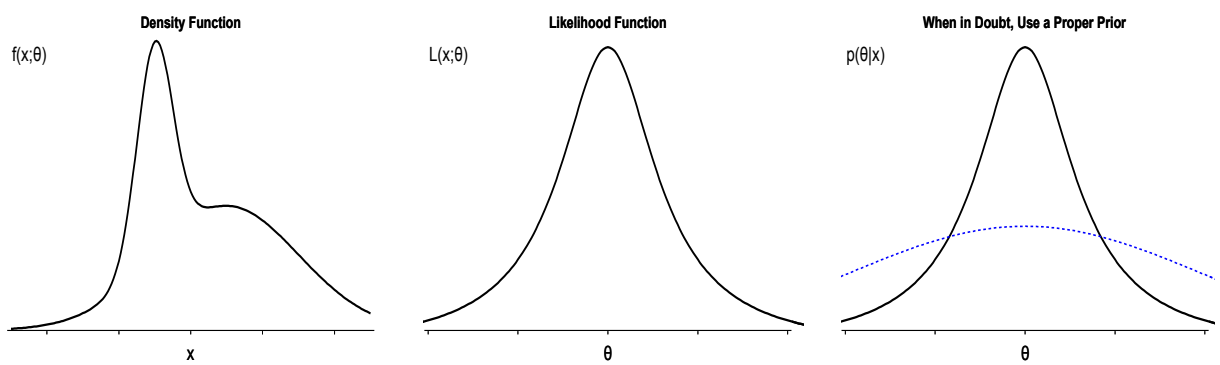


Flat Prior (Observation I)

If $\pi(\theta|\mathbf{x}) \propto \mathcal{L}(\theta)$ with $\pi(\theta) \propto 1$, then why not use the flat prior all the time?

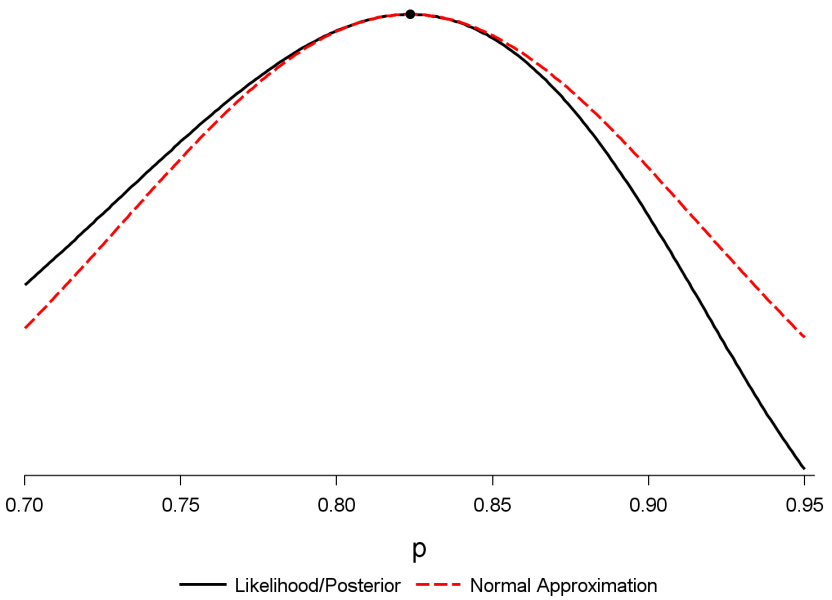
- Using a flat prior does not always guarantee a proper (integrable) posterior distribution; that is, $\int \pi(\theta|\mathbf{x})d\theta < \infty$.

The reason is that the likelihood function is only proper w.r.t. the random variable $\mathbf{X}$. But a posterior has to be integrable w.r.t. θ, a condition not required by the likelihood function.



Flat Prior (Observation II)

In cases where the likelihood function and the posterior distribution are identical, do we get the same answer?



Classical inference typically uses asymptotic results; Bayesian inference is based on exploring the entire distribution.

You Always Have to Defend Something!

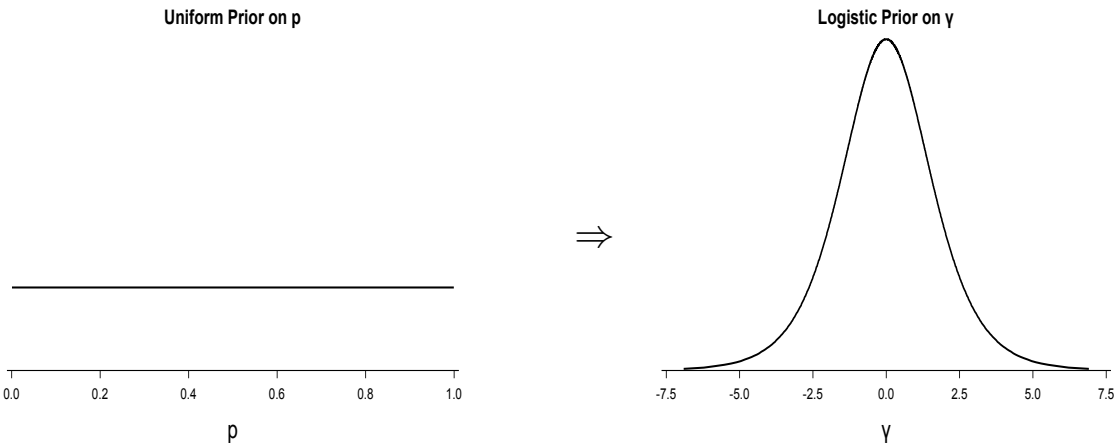
In a sense, everyone (Bayesian and non-Bayesian) is a slave to the likelihood function, which serves as a foundation to both paradigms. Given that,

- in Bayesian paradigm, you need to justify the selection of your prior
- in classical paradigm, you need to justify asymptotics: there exists an infinitely amount of unobserved data that are just like the ones that you have seen.

Flat Prior (Observation III)

Is flat prior noninformative? Suppose that, in the binomial example, you choose to model on $\gamma = \text{logit}(p)$ instead of p:

$$\pi(p) = \text{uniform}(0,1) \Leftrightarrow \pi(\gamma) = \text{logistic}(0,1)$$



You start with

$$\begin{aligned} p &= \frac{\exp(\gamma)}{1+\exp(\gamma)} = \frac{1}{1+\exp(-\gamma)} \\ \frac{\partial p}{\partial \gamma} &= -\frac{\exp(-\gamma)}{(1+\exp(-\gamma))^2} \end{aligned}$$

Do the transformation of variables, with the Jacobian:

$$\begin{aligned} \pi(p) &= 1 \cdot \mathbf{I}_{\{0 \le p \le 1\}} \\ \Rightarrow \pi(\gamma) &= \left|\frac{\partial p}{\partial \gamma}\right| \cdot \mathbf{I}_{\left\{0 \le \frac{1}{1+\exp(-\gamma)} \le 1\right\}} = \frac{\exp(-\gamma)}{(1+\exp(-\gamma))^2} \cdot \mathbf{I}_{\{-\infty \le \gamma \le \infty\}} \end{aligned}$$

The pdf for the logistic distribution with location a and scale b is

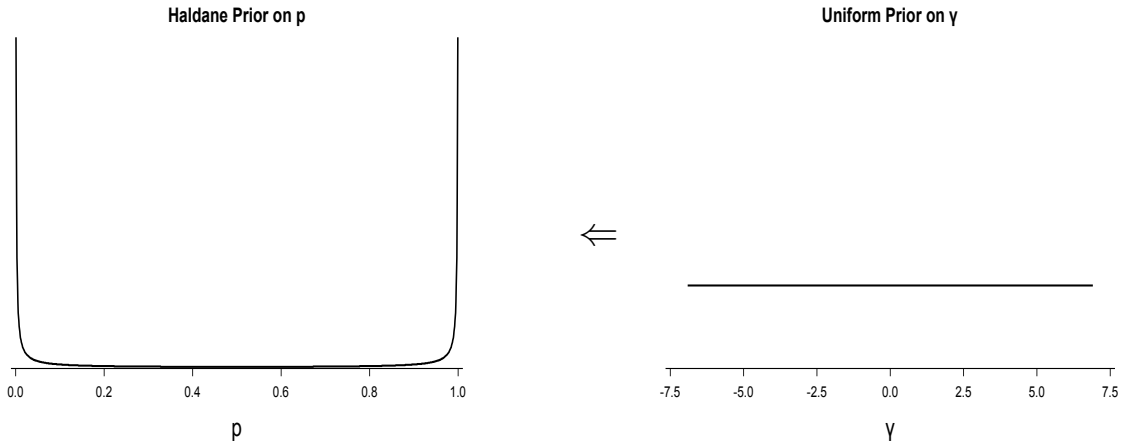
$$\exp\left(-\frac{\gamma-a}{b}\right) \Big/ b\left(1+\exp\left(-\frac{\gamma-a}{b}\right)\right)^2$$

and $\pi(\gamma) = \text{logistic}(0,1)$.

Flat Prior (Observation III)

If you choose to be noninformative on the γ dimension, you end up with a very different prior on the original p scale:

$$\pi(\gamma) \propto 1 \Leftrightarrow \pi(p) \propto p^{-1}(1-p)^{-1}$$



Flat Prior

- A flat prior implies a unit, a measurement scale, on which you assign equal likelihood
  - $\pi(\theta) \propto 1$: θ is as likely to be between (0, 1) as between (1000, 1001)
  - $\pi(\log(\theta)) \propto 1$ (equivalently, $\pi(\theta) \propto 1/\theta$): θ is as likely to be between (1, 10) as between (10, 100)
- One obvious difficulty in justifying a flat (uniform) prior is to explain the choice of unit which the prior is being noninformative on.
- Can we have a prior that is somewhat noninformative but at the same time is invariant to transformations?
  - Jeffreys' Prior

Jeffreys' Prior

Jeffreys' prior is defined as

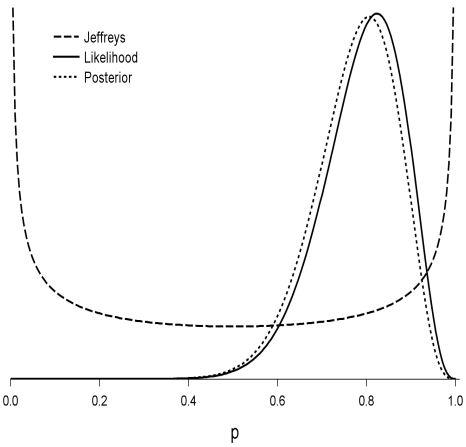
$$\pi(\theta) \propto |\mathbf{I}(\theta)|^{1/2}$$

where $|\cdot|$ denotes the determinant and $\mathbf{I}(\theta)$ is the expected Fisher information matrix based on the likelihood function $p(\mathbf{x}|\theta)$:

$$\mathbf{I}(\theta) = -E\left[\frac{\partial^2 \log p(\mathbf{x}|\theta)}{\partial \theta^2}\right]$$

In the Binomial Example:

$$\begin{aligned} \pi(p) &\propto p^{-1/2}(1-p)^{-1/2} \\ \mathcal{L}(p)\pi(p) &\propto p^{x-\frac{1}{2}}(1-p)^{n-x-\frac{1}{2}} \\ &\sim \text{Beta}(15.5, 4.5) \end{aligned}$$



Some Thoughts

Jeffreys' prior is

- *locally uniform*—a prior that does not change much over the region in which the likelihood is significant and does not assume large values outside that range. Hence it is somewhat noninformative.
- invariant with respect to one-to-one transformations.

The prior also

- can be improper for many models
- can be difficult to construct
- violates the likelihood principle

The Likelihood Principle

The likelihood principle states that, if two likelihood functions are proportional to each other,

$$L_1(\theta|\mathbf{x}) \propto L_2(\theta|\mathbf{x})$$

and one observes the same data $\mathbf{x}$, all inferences (about θ) should be the same.

Jeffreys' prior is in violation of this principle.

Negative Binomial Model

Instead of using a Binomial distribution, you can model the number of heads ($x = 14$) using a negative binomial distribution:

$$\mathcal{L}(q) = \binom{r+x-1}{x} q^r (1-q)^x$$

- x is the number of failures until $r = 3$ successes are observed
- q is the probability of success (getting a tail), and $1 - q$ is the probability of failure (getting a head)
- let $p = 1 - q$ and the likelihood function is rewritten as

$$\mathcal{L}(p) \propto (1-p)^r p^x$$

This is the same kernel as the binomial likelihood function.

Jeffreys' Prior

Same math leads to:

$$\frac{\partial^2 \ell p}{\partial p^2} = -\frac{x}{p^2} - \frac{r}{(1-p)^2}$$

Under a negative binomial model, $E(X) = \frac{r \cdot p}{1-p}$, and we have the following expected Fisher information:

$$\mathbf{I}(p) = \frac{-r}{p(1-p)^2}$$

The Jeffreys' prior becomes

$$\begin{aligned} \pi(p) &\propto p^{-1/2}(1-p)^{-1} \\ &\sim \text{Beta}\left(\frac{1}{2}, 0\right) \end{aligned}$$

A different prior, a different posterior, different inference on p.

The Cause

The cause to the problem is the expectation ($E(X)$), which depends on how the experiment is designed. In other words, taking the expectation means that we are making an assumption on how *all* future unobserved x behave.

Why do Bayesians consider this to be a problem?

- inference is based on yet-to-be-observed data and one might ended up being overly confident with the estimates.

Conjugate Prior

Conjugate prior is a family of prior distributions in which the prior and the posterior distributions are of the same family of distributions.

The Beta distribution is a conjugate prior to the binomial model:

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

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

The posterior distribution is also a Beta:

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

Conjugate Prior

$$\pi(p|\alpha,\beta,x,n) = \text{Beta}\,(x+\alpha, n-x+\beta)$$

One nice feature of the conjugate prior is that you can easily understand the amount information that is contained in the prior:

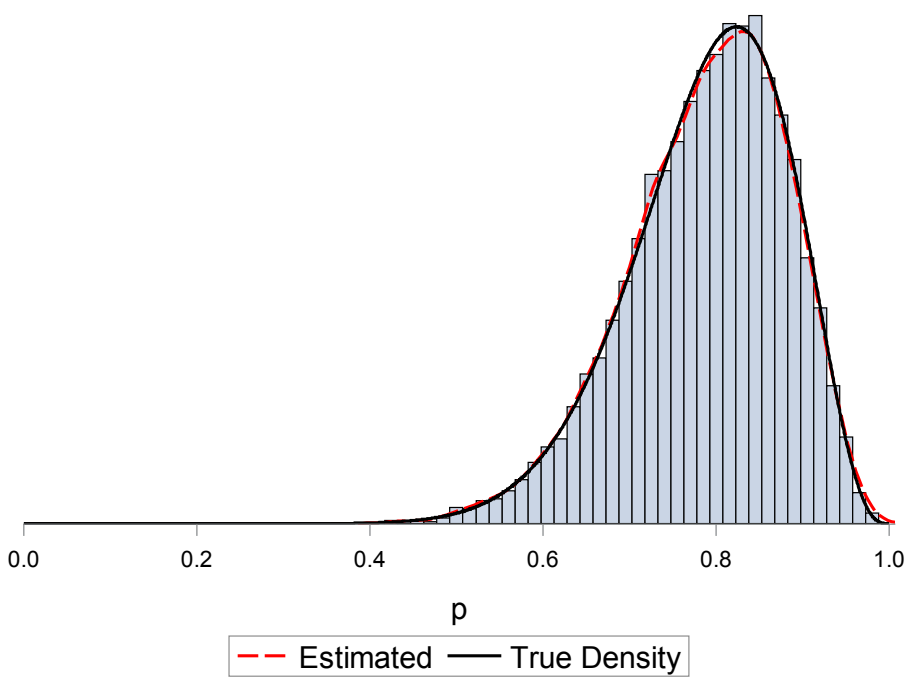
- the data contains x successes out of n trials
- the prior assumes α successes out of $\alpha+\beta$ trials: Beta$(2,2)$ clearly means different from Beta$(3,17)$

A related concept is the unit information (UI) prior (Kass and Wasserman (1995) JASA: 90:928-934), which is designed to contain roughly the same amount of information as one datum (variance equal to the inverse Fisher information based on one observation).

Bayesian Computation

- The key to Bayesian inferences is the posterior distribution
- Accurate estimation of the posterior distribution can be difficult and require a considerate amount of computation
- One of the most prevalent methods used nowadays is simulation-based:
  - repeatedly draw samples from a target distribution and use the collection of samples to empirically approximate the posterior

Simulation-based Estimation



How to do this for complex models that have many parameters?

Markov Chain Monte Carlo

- **Markov Chain**: a stochastic process that generates conditional independent samples according to some target distribution.
- **Monte Carlo**: a numerical integration technique that finds an expectation:

$$\mathrm{E}(f(\theta)) = \int f(\theta)p(\theta)d\theta \cong \frac{1}{n}\sum_{i=1}^{n} f(\theta_i)$$

with $\theta_1, \theta_2, \cdots, \theta_n$ being samples from $p(\theta)$.

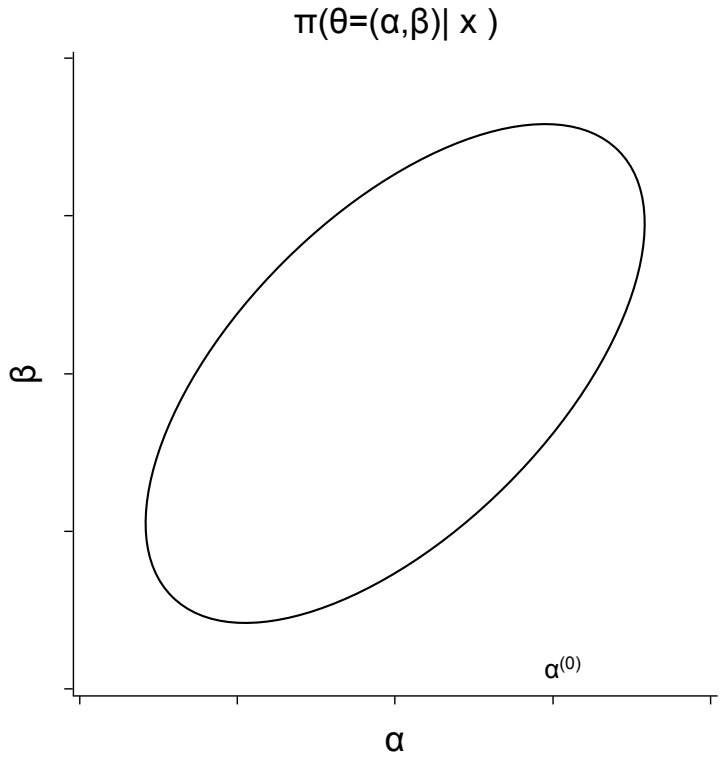
- **MCMC** is a method that generates a sequence of dependent samples from the target distribution and computes quantities by using Monte Carlo based on these samples.

Gibbs Sampler

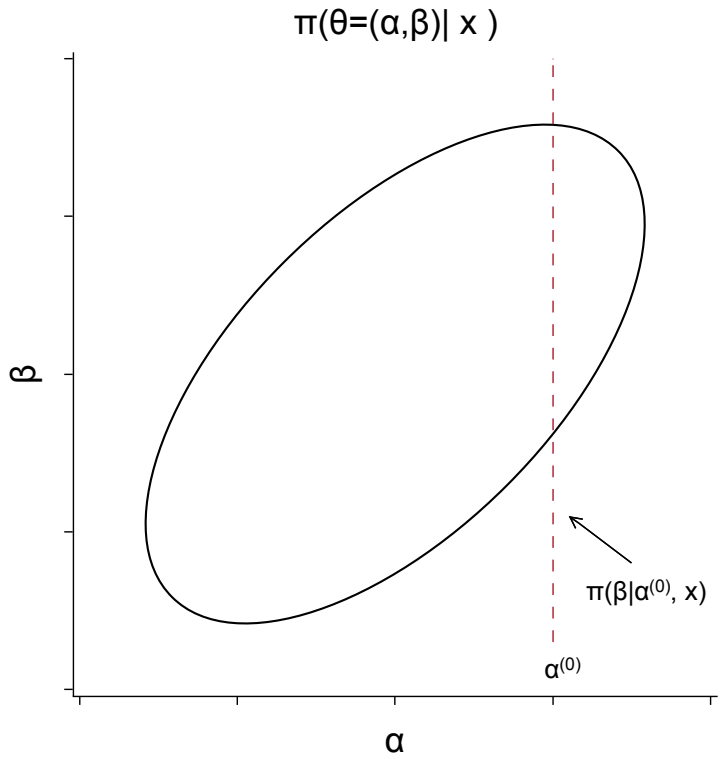
Gibbs sampler is an algorithm that sequentially generates samples from a joint distribution of two or more random variables. The sampler is often used when:

- The joint distribution, $\pi(\boldsymbol{\theta}|\mathbf{x})$, is not known explicitly
- The full conditional distribution of each parameter—for example, $\pi(\theta_i|\theta_j, i \neq j, \mathbf{x})$—is known

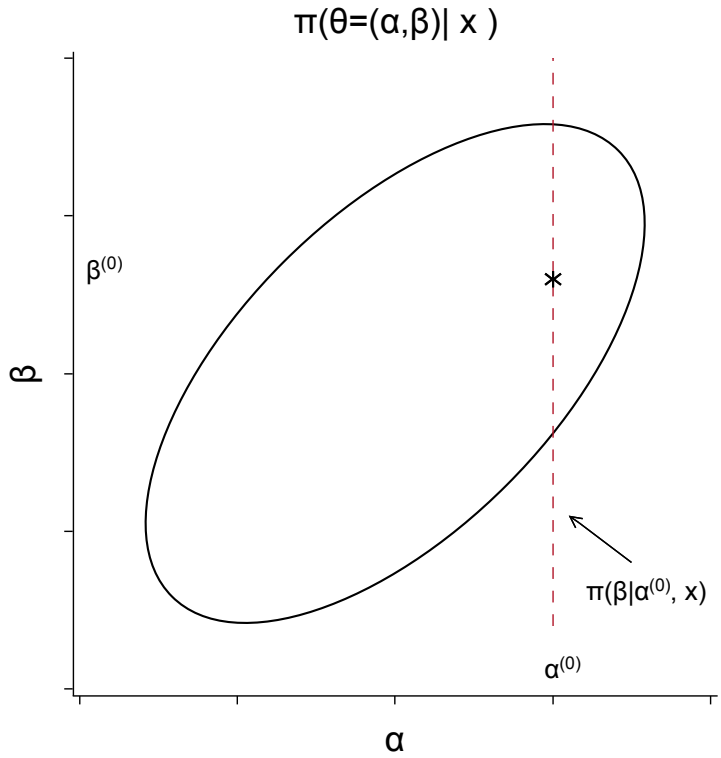
Gibbs Sampler



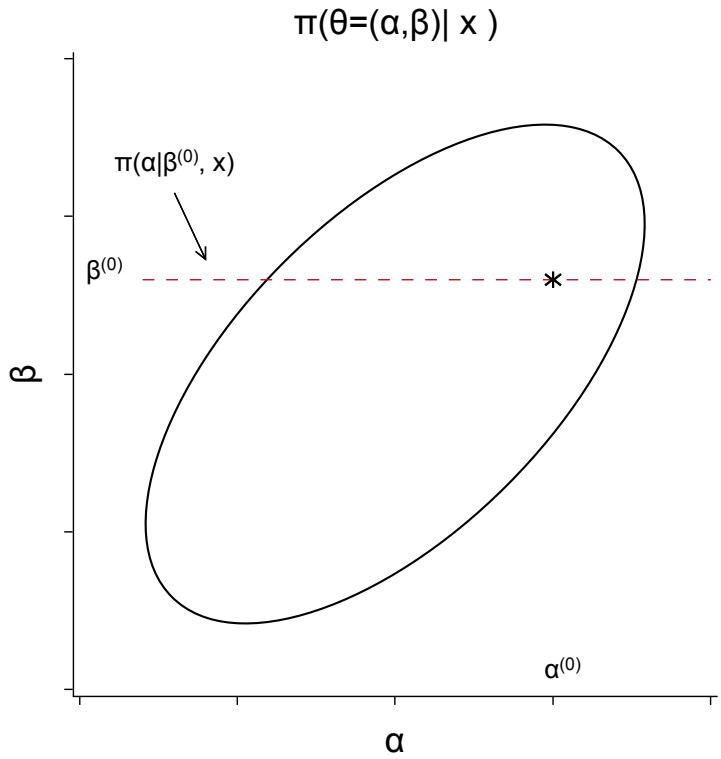
Gibbs Sampler



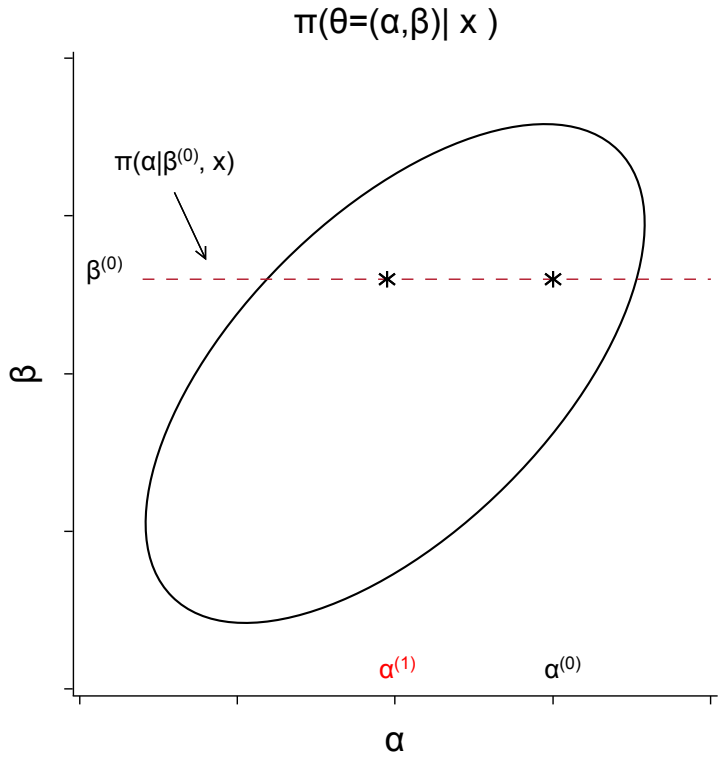
Gibbs Sampler



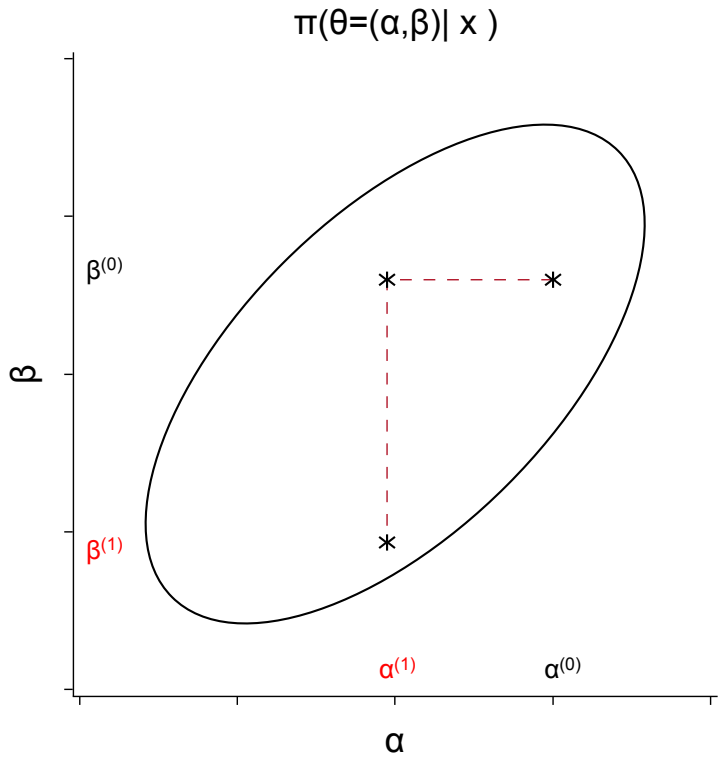
Gibbs Sampler



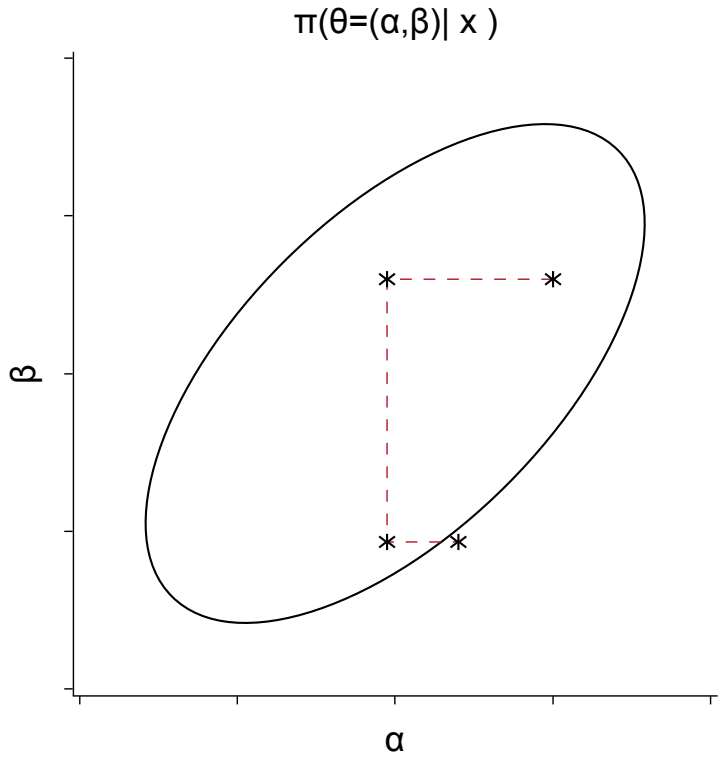
Gibbs Sampler



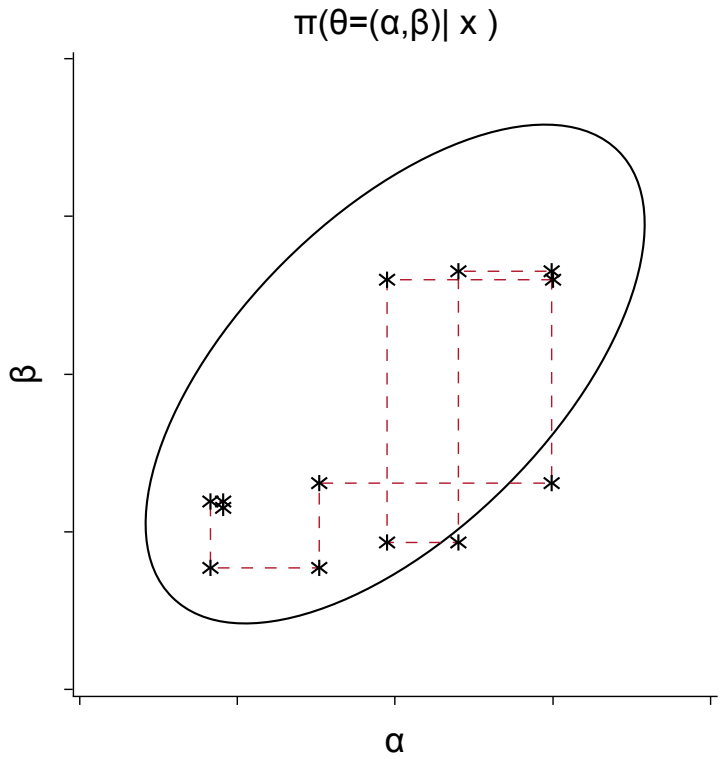
Gibbs Sampler



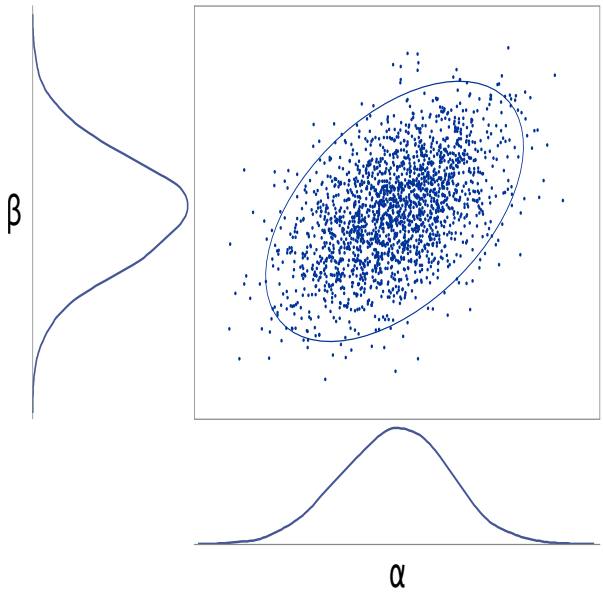
Gibbs Sampler



Gibbs Sampler



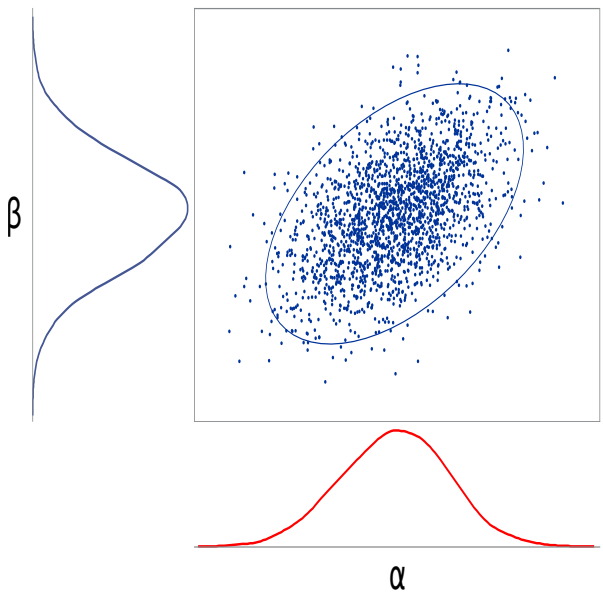
Joint and Marginal Distributions



$\pi(\alpha, \beta \mid \mathbf{x})$	
-1.1634	-1.0416
-1.0666	0.1060
4.0456	4.3485
-0.2531	0.9495
-0.4675	-0.8278
1.5945	2.5618
-3.1744	-4.5091
-0.3086	-2.6782
⋮	⋮

Gibbs enables you draw samples from a joint distribution.

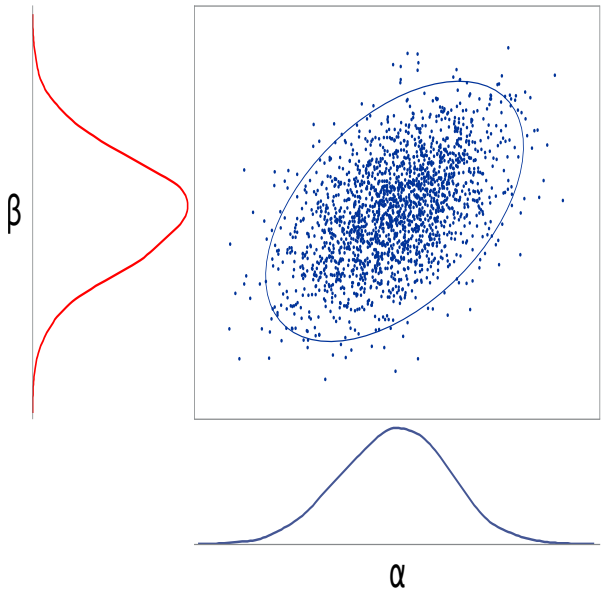
Joint and Marginal Distributions



$\pi(\alpha \mid \mathbf{x})$	
-1.1634	-1.0416
-1.0666	0.1060
4.0456	4.3485
-0.2531	0.9495
-0.4675	-0.8278
1.5945	2.5618
-3.1744	-4.5091
-0.3086	-2.6782
⋮	⋮

The by-products are the marginal distributions.

Joint and Marginal Distributions



	$\pi(\beta\|\mathbf{x})$
-1.1634	-1.0416
-1.0666	0.1060
4.0456	4.3485
-0.2531	0.9495
-0.4675	-0.8278
1.5945	2.5618
-3.1744	-4.5091
-0.3086	-2.6782
⋮	⋮

The by-products are the marginal distributions.

Gibbs Sampler

The difficulty in implementing a Gibbs sampler is how to efficiently generate from the conditional distribution, $\pi(\theta_i|\theta_j, i \neq j, \mathbf{x})$?

If each conditional distribution is a well known distribution, then it is easy.

Otherwise, you must use general algorithms to generate samples from a distribution:

- Metropolis Algorithm
- Adaptive Rejection Algorithm
- Slice Sampler
- ...

General algorithms typically have minimum requirements that are not distribution-specific, such as the ability to evaluate the objective functions.

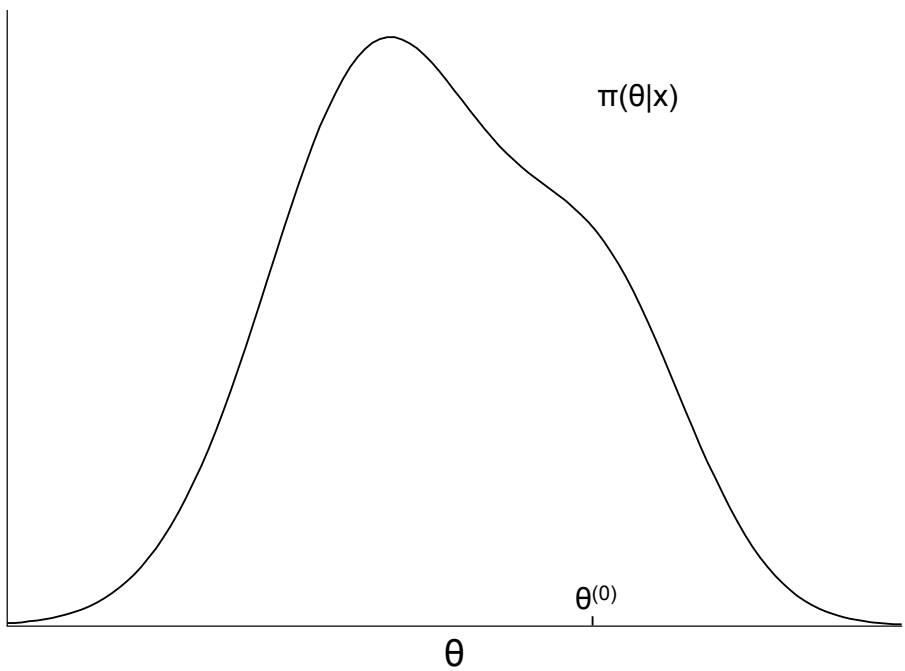
The Metropolis Algorithm

1. Let $t = 0$. Choose a starting point $\theta^{(t)}$. This can be an arbitrary point as long as $\pi(\theta^{(t)}|\mathbf{y}) > 0$.
2. Generate a new sample, θ', from a proposal distribution $q(\theta'|\theta^{(t)})$.
3. Calculate the following quantity:

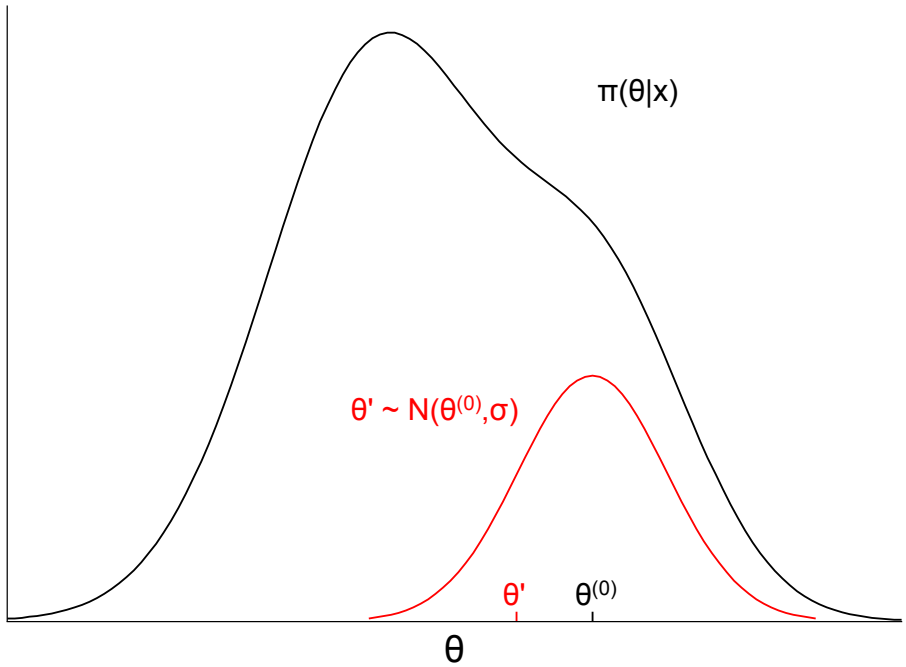
$$r = \min\left\{\frac{\pi(\theta'|\mathbf{y})}{\pi(\theta^{(t)}|\mathbf{y})}, 1\right\}$$

4. Sample u from the uniform distribution $U(0, 1)$.
5. Set $\theta^{(t+1)} = \theta'$ if $u < r$; $\theta^{(t+1)} = \theta^{(t)}$ otherwise.
6. Set $t = t + 1$. If $t < T$, the number of desired samples, go back to Step 2; otherwise, stop.

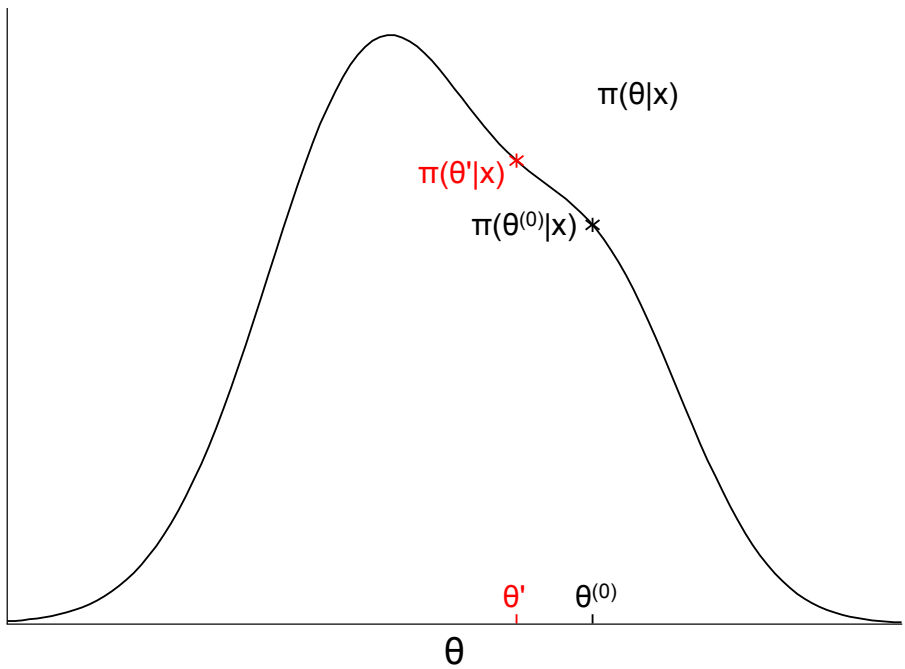
The Random-Walk Metropolis Algorithm



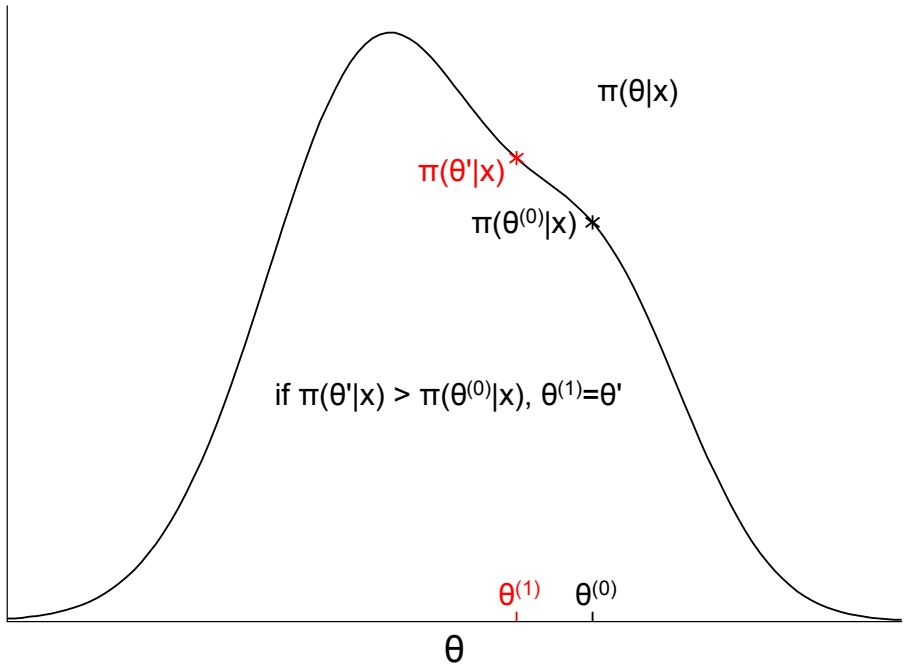
The Random-Walk Metropolis Algorithm



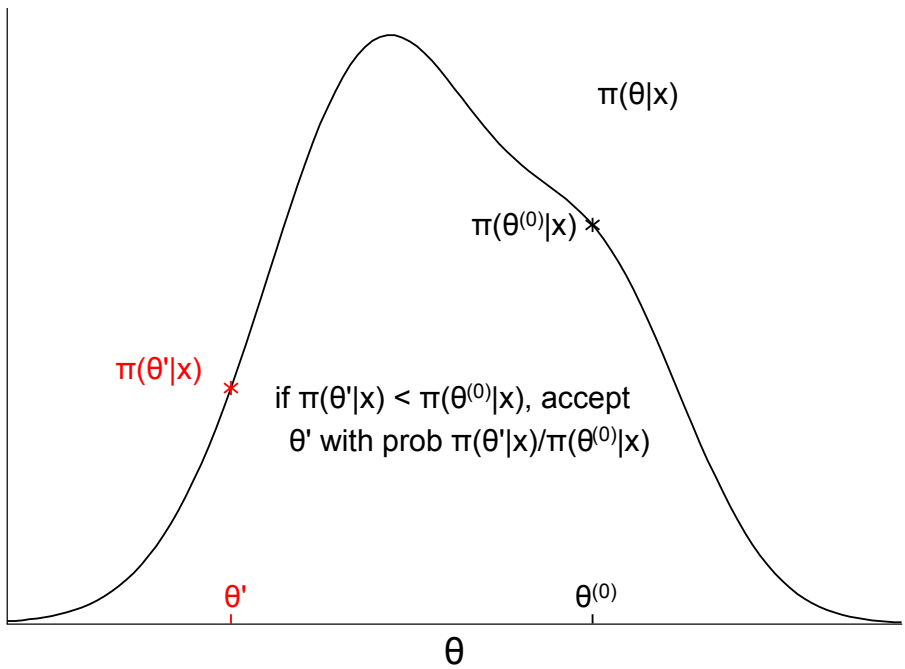
The Random-Walk Metropolis Algorithm



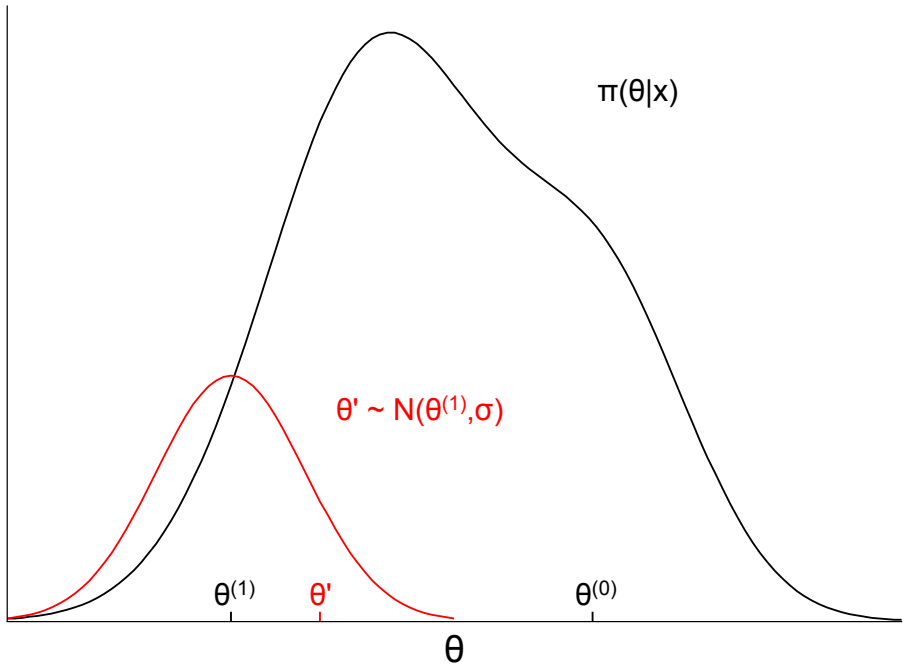
The Random-Walk Metropolis Algorithm



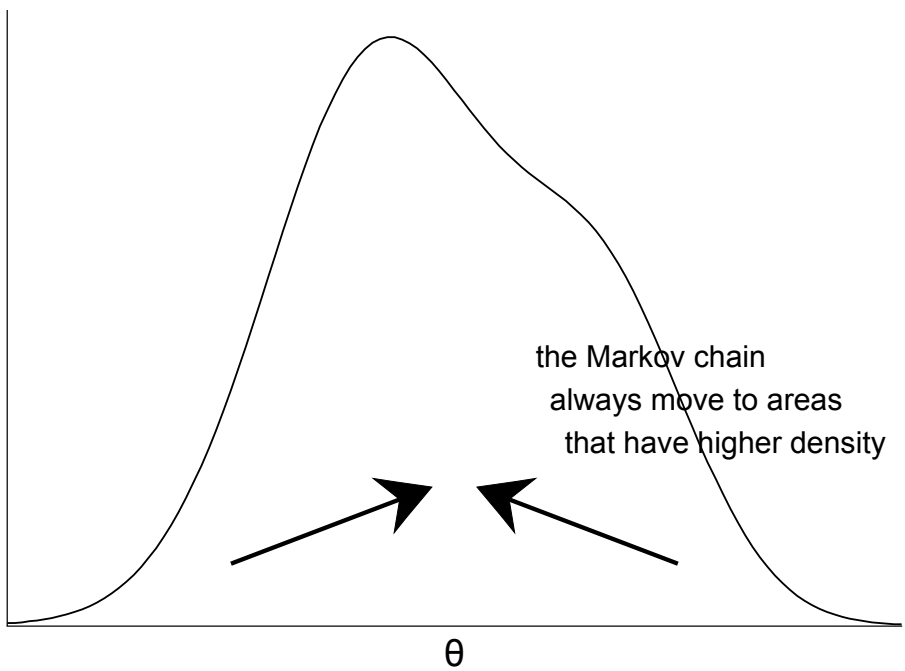
The Random-Walk Metropolis Algorithm



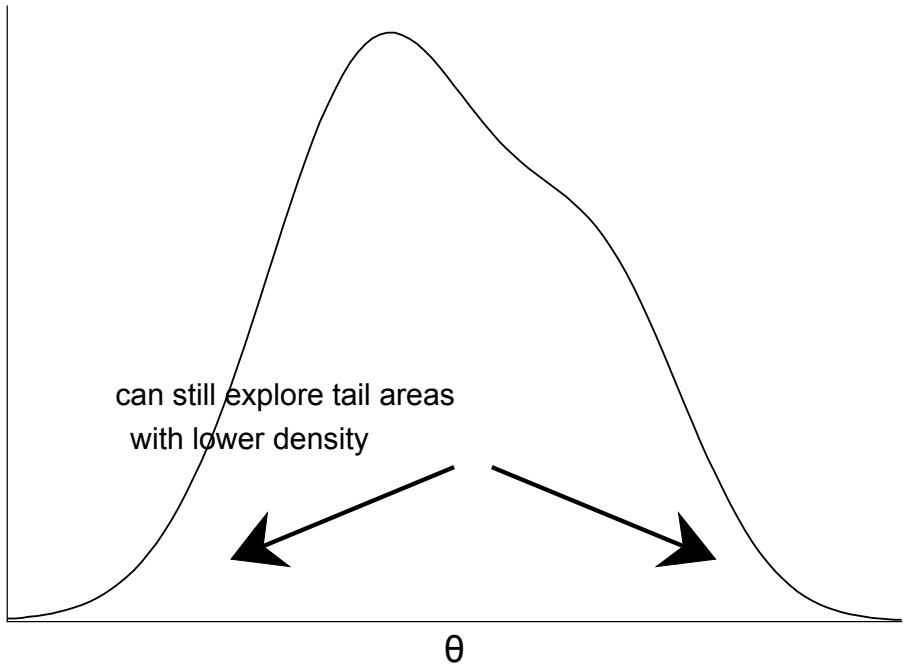
The Random-Walk Metropolis Algorithm



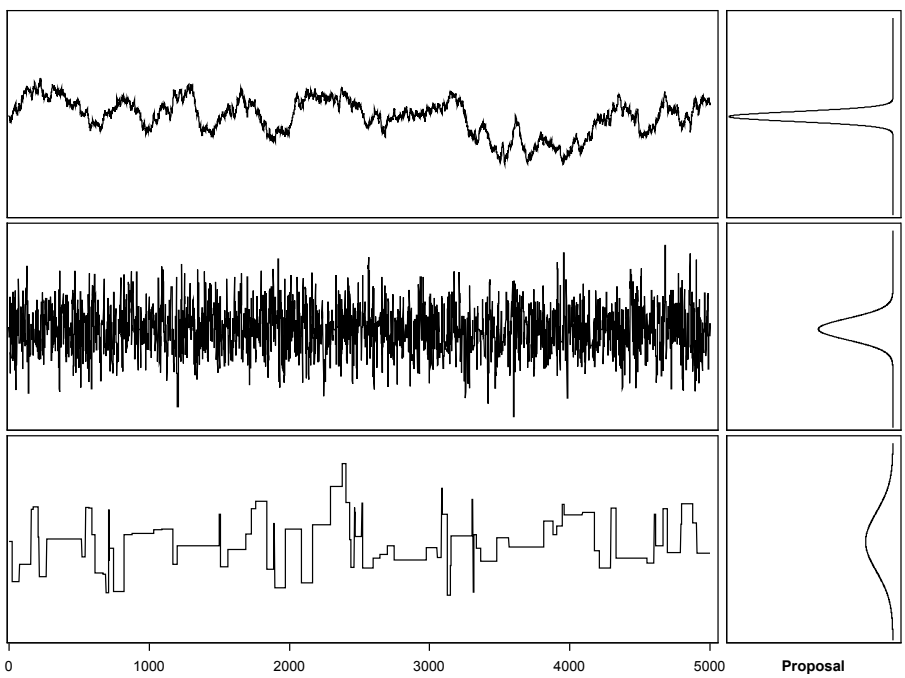
The Random-Walk Metropolis Algorithm



The Random-Walk Metropolis Algorithm



Scale and Mixing in the Metropolis



Markov Chain Convergence

An unconverged Markov chain does not explore the parameter space efficiently and the samples cannot approximate the target distribution well. Inference should not be based upon unconverged Markov chain, or very misleading results could be obtained.

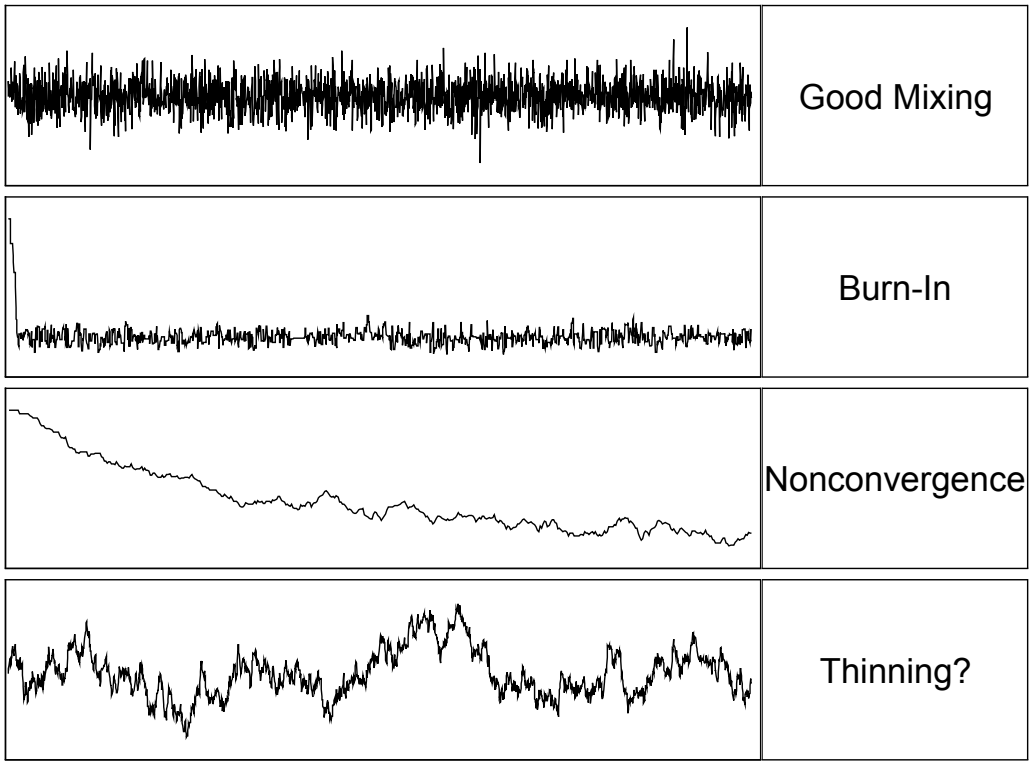
It is important to remember:

- Convergence should be checked for ALL parameters, and not just those of interest.
- There are no definitive tests of convergence. Diagnostics are often not sufficient for convergence.

Convergence Terminology

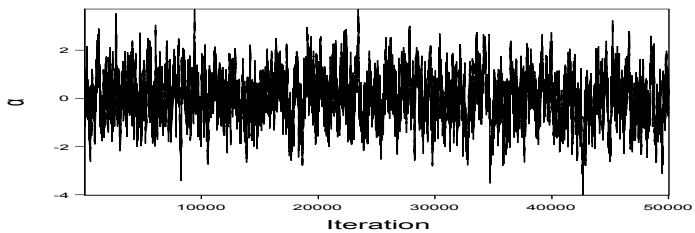
- **Convergence**: initial drift in the samples towards a stationary (target) distribution
- **Burn-in**: samples at start of the chain that are discarded to minimize their impact on the posterior inference
- **Slow mixing**: tendency for high autocorrelation in the samples. A slow-mixing chain does not traverse the parameter space efficiently.
- **Thinning**: the practice of collecting every kth iteration to reduce autocorrelation. Thinning a Markov chain can be wasteful because you are throwing away a $\frac{k-1}{k}$ fraction of all the posterior samples generated.
- **Trace plot**: plot of sampled values of a parameter versus iteration number.

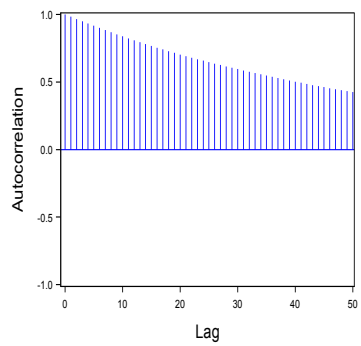
Various Trace Plots



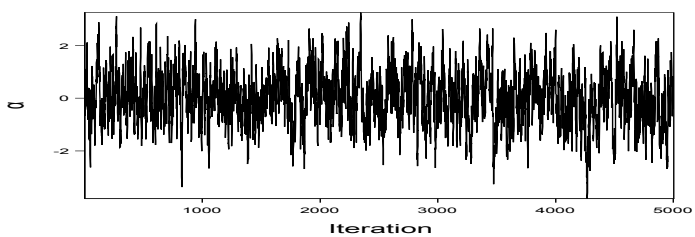
To Thin Or Not To Thin?

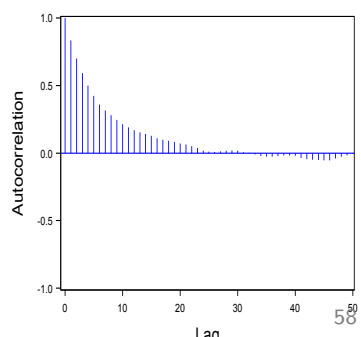
The argument for thinning is based on reducing autocorrelations, getting from





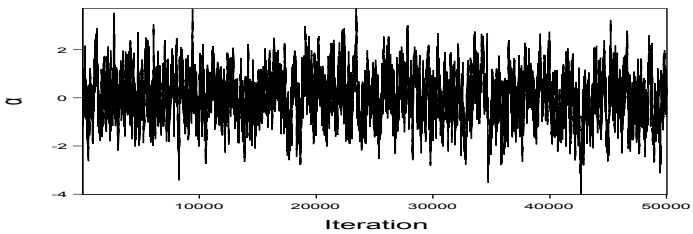
to

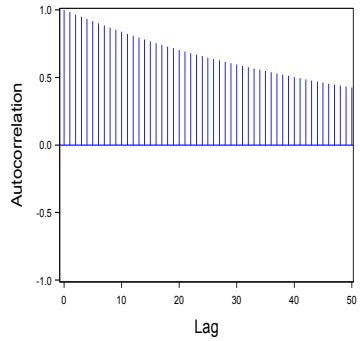




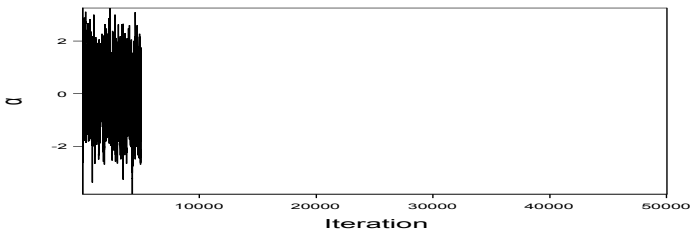
To Thin Or Not To Thin?

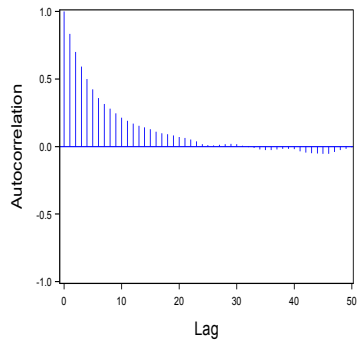
But at the same time, you are getting from





to





To Thin Or Not To Thin?

Thinning reduces autocorrelations and allows one to obtain seemingly independent samples. But at the same time, you throw away an appalling number of samples that can otherwise be used.

Autocorrelations do not lead to biased Monte Carlo estimates. It is simply an indicator of poor sampling efficiency.

On the other hand, sub-sampling loses information and actually increases the variance of sample mean estimators ($Var(\bar{\theta})$, not posterior variance). See MacEachern and Berliner (1994, *American Statistician*, 48:188).

Advice: unless storage becomes a problem, you are better off keeping all the samples for estimation.

Some Popular Convergence Diagnostics Tests

- Gelman-Rubin: tests whether multiple chains would convergent to the same target distribution.
- Geweke: tests whether the mean estimates have converged by comparing means from the early and latter part of the Markov chain.
- Heidelberger-Welch stationarity test: tests whether the Markov chain is a covariance (weakly) stationary process.
- Heidelberger-Welch halfwidth test: reports whether the sample size is adequate to meet the required accuracy for the mean estimate.
- Raftery-Lewis: evaluates the accuracy of the estimated (desired) percentiles by reporting the number of samples needed to reach the desired accuracy of the percentiles.

More on Convergence Diagnosis

There are no definitive tests of convergence.

- *With experience*, visual inspection of trace plots is often the most useful approach.
- Geweke and Heidelberger-Welch sometimes reject even when the trace plots look good.
- Oversensitivity to minor departures from stationarity does not impact inferences.
- Different convergence diagnostics are designed to protect you against different potential pitfalls.
- ESS is frequently a good numerical indicator on the status of mixing.

Effective Sample Size (ESS)

ESS (Kass et al. 1998, *American Statistician*, 52:93) provides a measure on how well a Markov chain is mixing.

$$\mathrm{ESS} = \frac{n}{1 + 2\sum_{k=1}^{(n-1)} \rho_k(\theta)}$$

where n is the total sample size and $\rho_k(\theta)$ is the autocorrelation of lag k for θ.

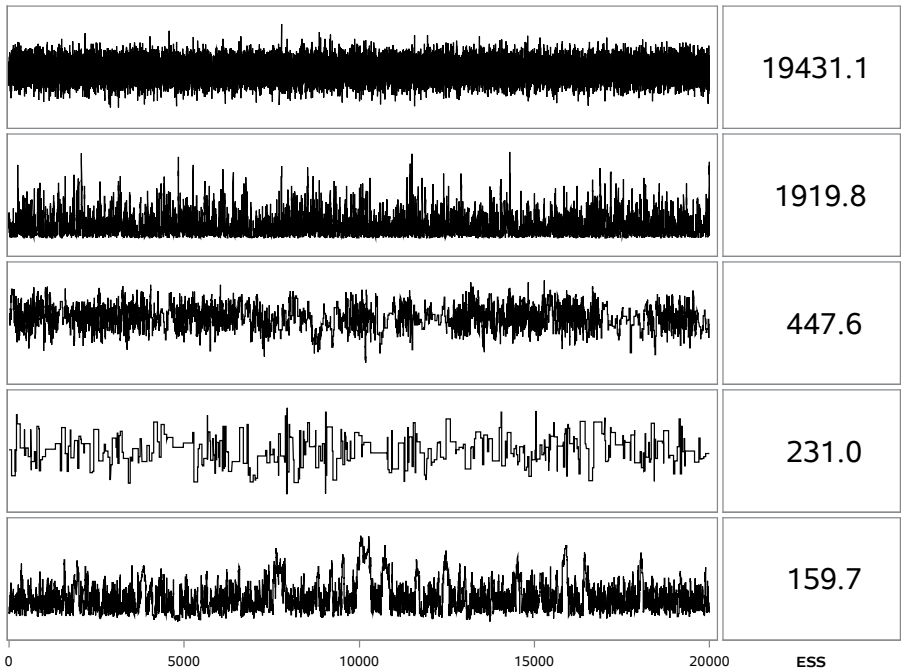
- The closer ESS is to n, the better mixing is in the Markov chain.
- ESS of size around 1,000 is mostly sufficient in estimating the posterior density. You want increase the number for tail percentiles.

Effective Sample Size (ESS)

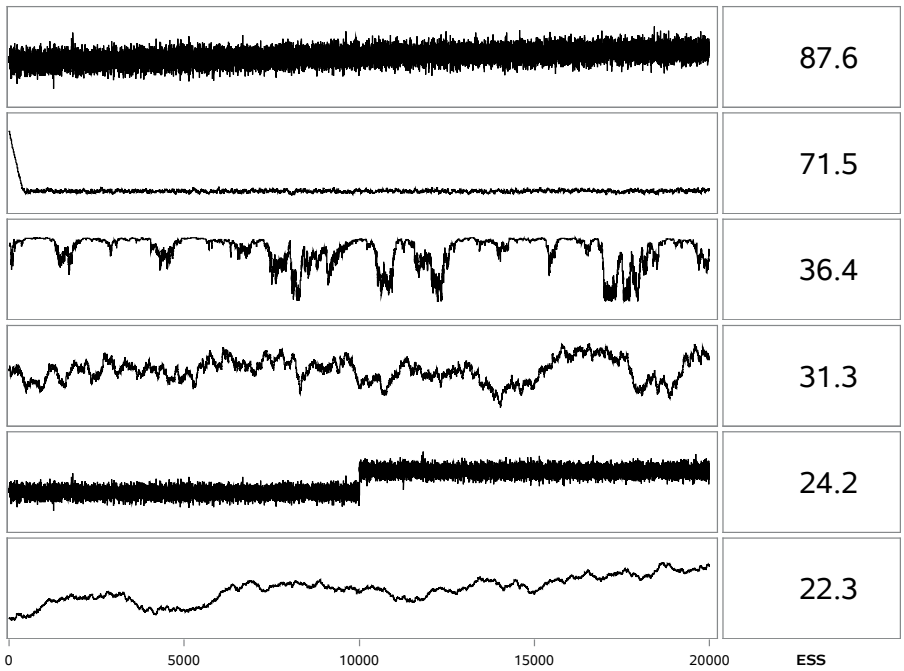
I personally prefer to use ESS as a way to judge convergence:

- small numbers of ESSs often indicate "something isn't quite right."
- large numbers of ESSs are typically good news
- moves away from the conundrum of dealing with and interpreting hypothesis testing results
- You can summarizes the convergence of multiple parameters by looking at the distribution of all the ESSs, or even the minimum ESS (worst case).

Various Trace Plots and ESSs



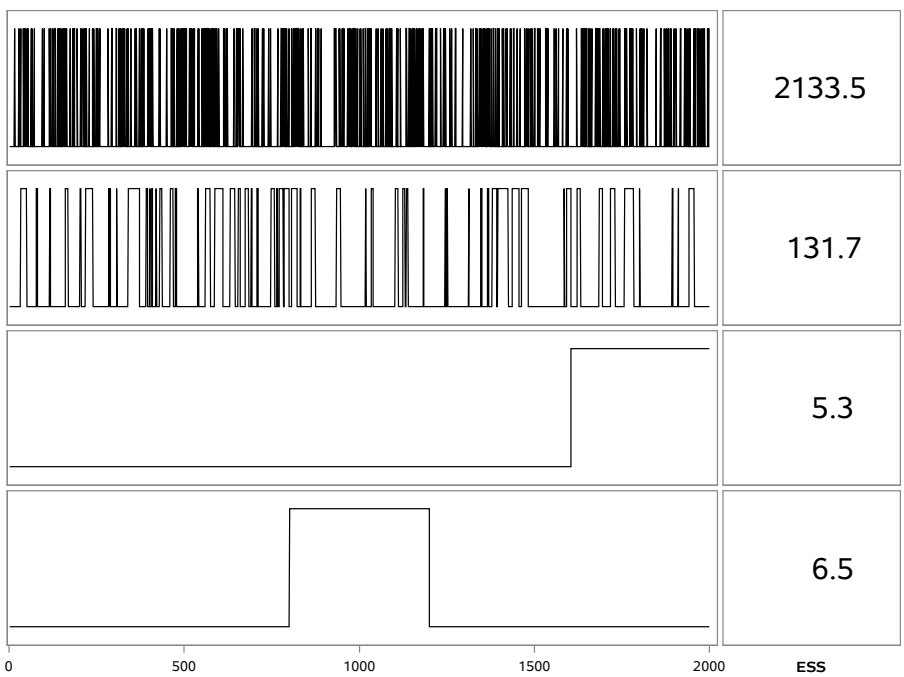
Various Trace Plots and ESSs



More on ESS

- ESS is not significance test-based, and you can think of it as more of a numerical criterion, similar to convergence criteria used in optimizations.
- You can still get good ESSs in "unconverged" chains, such as a chain that is stuck in a local mode in a multi-mode problem.
  - These are fairly rare (and often there are plenty of other signs to indicate such complex problems).
- Bad ESSs serves as a good indicator when things go bad
  - problems can sometimes be easily corrected (burn-in, longer chain, etc).
  - false rejections (bad ESSs from convergened chains) are less common, but do exist (in binary and discrete parameters).

Bernoulli Markov Chains, all with Marginal Prob of 0.2



Outline of Part II

- Overview of Bayesian capabilities in the GENMOD, PHREG, LIFEREG, and FMM procedures
- Overview of the BAYES statement and syntax for requesting Bayesian analysis
- Examples
  - GENMOD: linear regression
  - GENMOD: Poisson regression
  - PHREG: Cox model
  - PHREG: piecewise exponential model (optional)

The GENMOD, PHREG, LIFEREG, and FMM Procedures

These four procedures provide:

- The BAYES statement
- A set of frequently used prior distributions (noninformative, Jeffreys'), posterior summary statistics, and convergence diagnostics
- Various sampling algorithms: conjugate, direct, adaptive rejection (Gilks and Wild 1992; Gilks, Best, and Tan 1995), Metropolis, Gamerman algorithm, etc.

Bayesian capabilities include:

- GENMOD: Generalized Linear Models
- LIFEREG: Parametric Lifetime Models
- PHREG: Cox Regression (Frailty) and Piecewise Exponential Models
- FMM: Finite Mixture Models

Prior Distributions in SAS Procedures

- *Uniform* (or *flat*)prior is defined as:

$$\pi(\theta) \propto 1$$

  This prior is not integrable, but it does not lead to improper posterior in any of the procedures.
- *Improper* prior is defined as:

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

  This prior is often used as a noninformative prior on the scale parameter, and it is uniform on the log-scale.
- *Proper* prior distributions include gamma, inverse-gamma, AR(1)-gamma, normal, multivariate normal densities.
- *Jeffreys'* prior is provided in PROC GENMOD.

Syntax for the BAYES Statement

The BAYES statement is used to request all Bayesian analysis in these procedures.

BAYES < options > ;

The following options appear in all BAYES statements:

INITIAL=	initial values of the chain
NBI=	number of burn-in iterations
NMC=	number of iterations after burn-in
OUTPOST=	output data set for posterior samples
SEED=	random number generator seed
THINNING=	thinning of the Markov chain
DIAGNOSTICS=	convergence diagnostics
PLOTS=	diagnostic plots
SUMMARY=	summary statistics
COEFFPRIOR=	prior for the regression coefficients

Regression Example

Consider the model

$$Y = \beta_0 + \beta_1 LogX1 + \epsilon$$

where Y is the survival time, $LogX_1$ is log(blood-clotting score), and ϵ is a $N(0, \sigma^2)$ error term.

The default priors that PROC GENMOD uses are:

$$\begin{aligned} \pi(\beta_0) &\propto 1 \quad \pi(\beta_1) \propto 1 \\ \pi(\sigma^2) &\sim \text{gamma}(\text{shape} = 2.001, \text{iscale} = 0.0001) \end{aligned}$$

Regression Example

A subset of the data and statements fit Bayeisna regression:

```
data surg;
   input logy logx1 @@;
   datalines;
   199.986    1.90211   100.995    1.62924   203.986    2.00148
   100.995    1.87180   508.979    2.05412    80.002    1.75786
   ...
   ;
proc genmod data=surg;
   model y = logx1  / dist=normal link=identity;
   bayes seed=4 outpost=post diagnostics=all summary=all;
run;
```

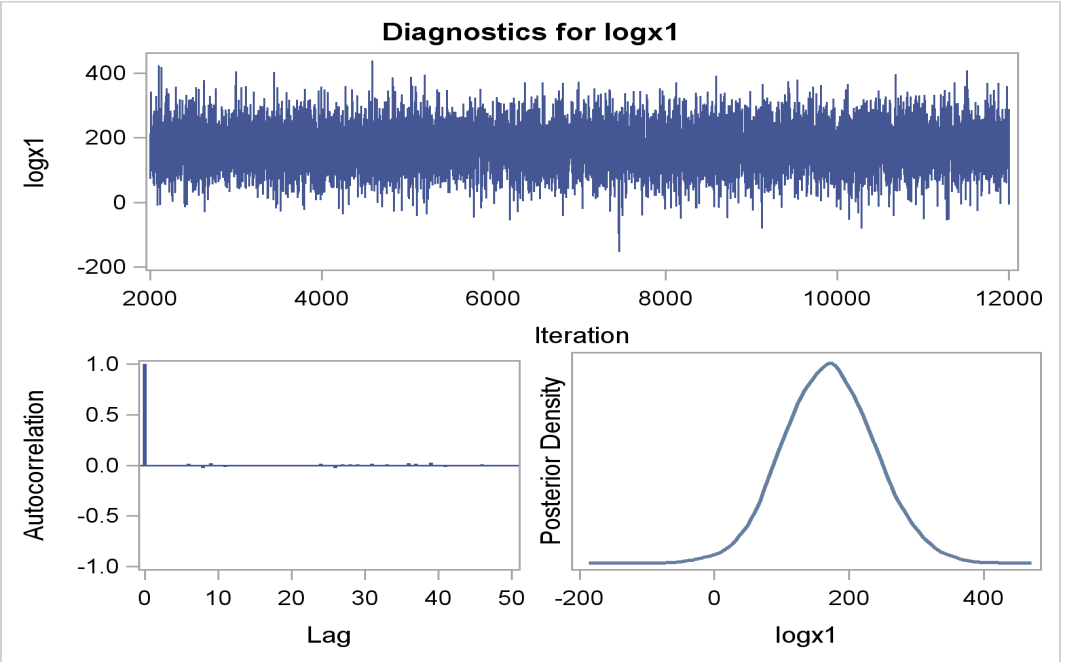
SEED specifies a random seed

OUTPOST saves posterior samples

DIAGNOSTICS requests all convergence diagnostics

SUMMARY requests calculation for all posterior summary statistics

Convergence Diagnostics for β_1



Mixing

The following are the autocorrelation and effective sample sizes. The mixing appears to be very good, which agrees with the trace plots.

Bayesian Analysis

Posterior Autocorrelations				
Parameter	**Lag 1**	**Lag 5**	**Lag 10**	**Lag 50**
Intercept	0.0062	0.0105	0.0244	-0.0003
logx1	0.0045	0.0106	0.0269	0.0009
Dispersion	-0.0077	0.0116	0.0082	-0.0003

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

Additional Convergence Diagnostics

Bayesian Analysis

Gelman-Rubin Diagnostics		
Parameter	Estimate	97.5% Bound
Intercept	1.0000	1.0002
logx1	1.0000	1.0002
Dispersion	0.9999	0.9999

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	2	3789	3746	1.0115
logx1	2	3834	3746	1.0235
Dispersion	.	.	3746	.

Bayesian Analysis

Geweke Diagnostics		
Parameter	z	Pr > |z|
Intercept	1.0623	0.2881
logx1	-1.0554	0.2912
Dispersion	0.6388	0.5229

Heidelberger-Welch Diagnostics								
	Stationarity Test				Half-width Test			
Parameter	Cramer-von-Mises Stat	p	Test Outcome	Iterations Discarded	Half-width	Mean	Relative Half-width	Test Outcome
Intercept	0.0587	0.8223	Passed	0	2.3604	-94.5279	-0.0250	Passed
logx1	0.0611	0.8069	Passed	0	1.4139	169.9	0.00832	Passed
Dispersion	0.1055	0.5585	Passed	0	67.9392	18478.4	0.00368	Passed

Summarize Convergence Diagnostics

- **Autocorrelation**: shows low dependency among Markov chain samples
- **ESS**: values close to the sample size indicate good mixing
- **Gelman-Rubin**: values close to 1 suggest convergence from different starting values
- **Geweke**: indicates mean estimates are stabilized
- **Raftery-Lewis:** shows sufficient samples to estimate 0.025 percentile within $+/-$ 0.005 accuracy
- **Heidelberger-Welch**: suggests the chain has reached stationarity and there are enough samples to estimate the mean accurately

Posterior Summary and Interval Estimates

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
Intercept	10000	-94.5279	119.1	-172.9	-95.0444	-16.1862
logx1	10000	169.9	68.5847	124.6	170.2	214.4
Dispersion	10000	18478.4	3670.4	15825.8	17987.2	20596.5

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	-331.1	140.6	-320.5	147.5
logx1	0.050	35.3949	306.1	35.3925	306.1
Dispersion	0.050	12646.1	27050.8	11957.9	25806.1

Posterior Inference

Posterior correlation:

Bayesian Analysis

Posterior Correlation Matrix			
Parameter	**Intercept**	**logx1**	**Dispersion**
Intercept	1.000	-0.987	-0.007
logx1	-0.987	1.000	0.006
Dispersion	-0.007	0.006	1.000

Fit Statistics

PROC GENMOD also calculates the Deviance Information Criterion (DIC)

Bayesian Analysis

Fit Statistics	
DIC (smaller is better)	690.182
pD (effective number of parameters)	3.266

Posterior Probabilities

Suppose that you are interested in knowing whether $LogX1$ has a positive effect on survival time. Quantifying that measurement, you can calculate the probability $\beta_1 > 0$, which can be estimated directly from the posterior samples:

$$Pr(\beta_1 > 0 | Y, LogX1) = \frac{1}{N} \sum_{t=1}^{N} I(\beta_1^t > 0)$$

where $I(\beta_1^t > 0) = 1$ if $\beta_1^t > 0$ and 0 otherwise. $N = 10,000$ is the sample size in this example.

Posterior Probabilities

The following SAS statements calculate the posterior probability:

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

ods select summary;
proc means data = Prob(keep=Indicator) n mean;
run;
```

The probability is roughly 0.9926, which strongly suggests that the slope coefficient is greater than 0.

Outline

Binomial model

Consider a study of the analgesic effects of treatments on elderly patients with neuralgia.

- Two test treatements and a placebo are compared.
- The response variable is whether the patient reported pain or not.
- Covariates include the age and gender of 60 patients and the duration of complaint before the treatment began.

The Data

A subset of 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
...
P  M  67  17  Yes  B  M  70  22  No  A  M  65  15  No
P  F  67   1  Yes  A  M  67  10  No  P  F  72  11  Yes
A  F  74   1  No   B  M  80  21  Yes A  F  69   3  No
;
```

Treatment: A, B, P

Sex: F, M

Pain: Yes, No

The Model

A logistic regression is considered for this data set:

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

where Sex_F, $\text{Treatment}_\text{A}$, and $\text{Treatment}_\text{B}$ are dummy variables for the categorical predictors.

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

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

Logistic Regression

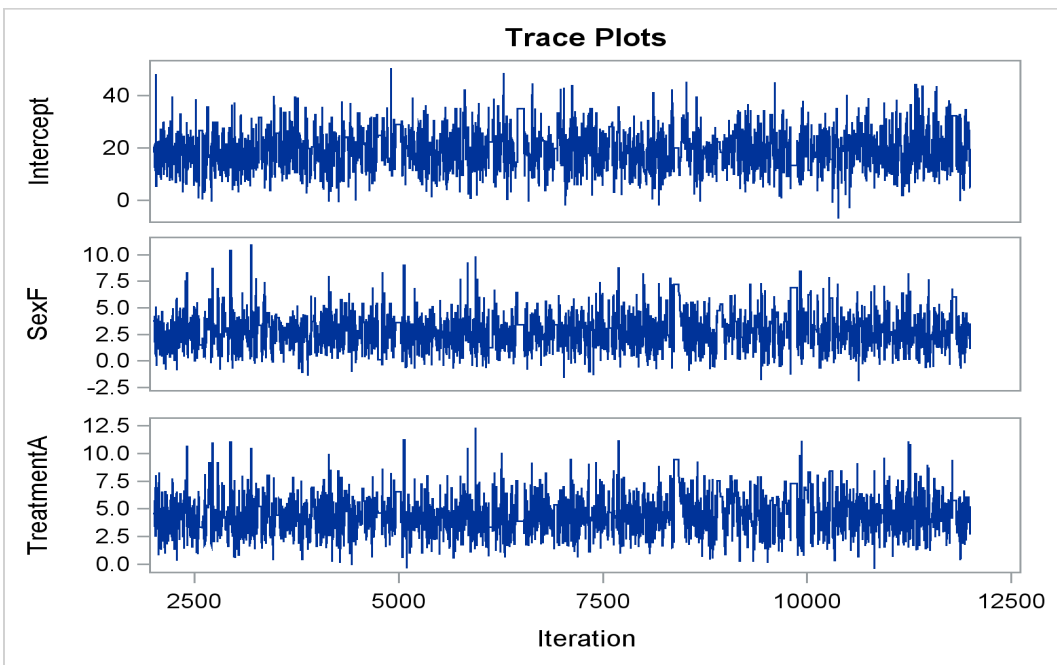
The following statements fit a Bayesian logistic regression model in PROC GENMOD:

```
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;
run;
```

- PROC GENMOD models the probability of *no pain* (`Pain` = No)
- The default sampling algorithm is the Gamerman algorithm (Gamerman, D. 1997, *Statistics and Computing*, 7:57). PROC GENMOD offers a couple of alternative sampling algorithms, such as adaptive rejection and independence Metropolis.

Logistic Regression

Trace plots of some of the parameters.



Logistic Regression

Posterior summary statistics:

Bayesian Analysis

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
Intercept	10000	19.5936	7.7544	13.9757	19.0831	24.7758
SexF	10000	2.9148	1.7137	1.7348	2.8056	3.9222
TreatmentA	10000	4.6190	1.7924	3.3880	4.4333	5.6978
TreatmentB	10000	5.1406	1.8808	3.7928	5.0154	6.2784
TreatmentASexF	10000	-1.0367	2.3097	-2.4499	-0.9233	0.4706
TreatmentBSexF	10000	-0.3478	2.2499	-1.7787	-0.3578	1.1129
Age	10000	-0.3372	0.1155	-0.4141	-0.3276	-0.2531
Duration	10000	0.00894	0.0366	-0.0160	0.00926	0.0328

Logistic Regression

Posterior interval statistics:

Bayesian Analysis

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	5.9732	35.0404	6.7379	35.5312
SexF	0.050	0.0155	6.9155	-0.1694	6.5110
TreatmentA	0.050	1.5743	8.4046	1.4277	8.0465
TreatmentB	0.050	1.7895	9.0056	2.0476	9.0766
TreatmentASexF	0.050	-5.6692	3.4066	-5.6793	3.2184
TreatmentBSexF	0.050	-4.7148	4.2324	-4.9417	3.8466
Age	0.050	-0.5724	-0.1325	-0.5735	-0.1372
Duration	0.050	-0.0626	0.0836	-0.0628	0.0799

Odds Ratio

In the logistic model, the log odds function, logit(X), is given by:

$$\text{logit}(X) \equiv \log\left(\frac{\Pr(Y=1 \mid X)}{\Pr(Y=0 \mid X)}\right) = \beta_0 + X\beta_1$$

Suppose that you are interested in calculating the ratio of the odds for the female patients ($\text{Sex}_\text{F} = 1$) to the male patients ($\text{Sex}_\text{F} = 0$). The log of the odds ratio is the following:

$$\begin{aligned}
\log(\psi) &\equiv \log(\psi(\text{Sex}_\text{F} = 1, \text{Sex}_\text{F} = 0)) \\
&= \text{logit}(\text{Sex}_\text{F} = 1) - \text{logit}(\text{Sex}_\text{F} = 0) \\
&= (\beta_0 + 1 \times \beta_1) - (\beta_0 + 0 \times \beta_1) \\
&= \beta_1
\end{aligned}$$

It follows that the odds ratio is:

$$\psi = \exp(\beta_1)$$

Odds Ratio

Note that, by default, PROC GENMOD uses PARAM=GLM parametrization, which codes 1 and -1 to the values of Sex_F.

In general, suppose the values of Sex_F are coded as constants a and b instead of 0 and 1.

- The odds when $\text{Sex}_\text{F} = a$ become $\exp(\beta_0 + a \cdot \beta_1)$
- The odds when $\text{Sex}_\text{F} = b$ become $\exp(\alpha + b \cdot \beta_1)$

The odds ratio is

$$\psi = \exp[(b-a)\beta_1] = [\exp(\beta_1)]^{b-a}$$

In other words, for any types of the effect parametrization schemes, as long as $b - a = 1$, $\psi = \exp(\beta_1)$

Odds Ratio

Odds ratios are functions of the model parameters, which can be obtained by manipulating posterior samples generated by PROC GENMOD. To estimate posterior odds ratios,

- save PROC GENMOD analysis to a SAS item store
- postfit odds ratios using the ESTIMATE statement in PROC PLM

An item store is a special SAS-defined binary file format used to store and restore information with a hierarchical structure.

The PLM procedure performs postprocessing tasks by taking the posterior samples (from GENMOD) and estimate functions of interest.

The ESTIMATE statement provides a mechnism for obtaining custom hypothesis testing (or linear combination of the regression coefficients).

Odds Ratio

The following statements fit the model in PROC GENMOD and saves the content to a SAS item store (`logit_bayes`):

```
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;
   store logit_bayes;
run;
```

Odds Ratio

The following statements evoke PROC PLM and estimate the odds ratio between the female group and male group conditional on treatment A:

```
proc plm restore=logit_bayes;
   estimate "F vs M, at Trt=A"
        sex 1 -1 treatment*sex [1, 1 1] [-1, 1 2]
         / e exp cl plots=dist;
   run;
```

sex 1 -1 : estimates the difference between β_1 and β_2, which under the GLM parametrization, is equal to β_1

treatment * sex ... : assigns **1** to the interation where "treatment=1" and "sex=1", and **-1** to the interaction where "treatment=1" and "sex=2"

e : requests that the **L** matrix coefficients be displayed

exp : exponentials and displays estimates (exp β_1)

cl : constructs 95% credit intervals

plots : generates histograms with kernel density overlaid

L Matrix Coefficients (GLM Parametrization)

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			

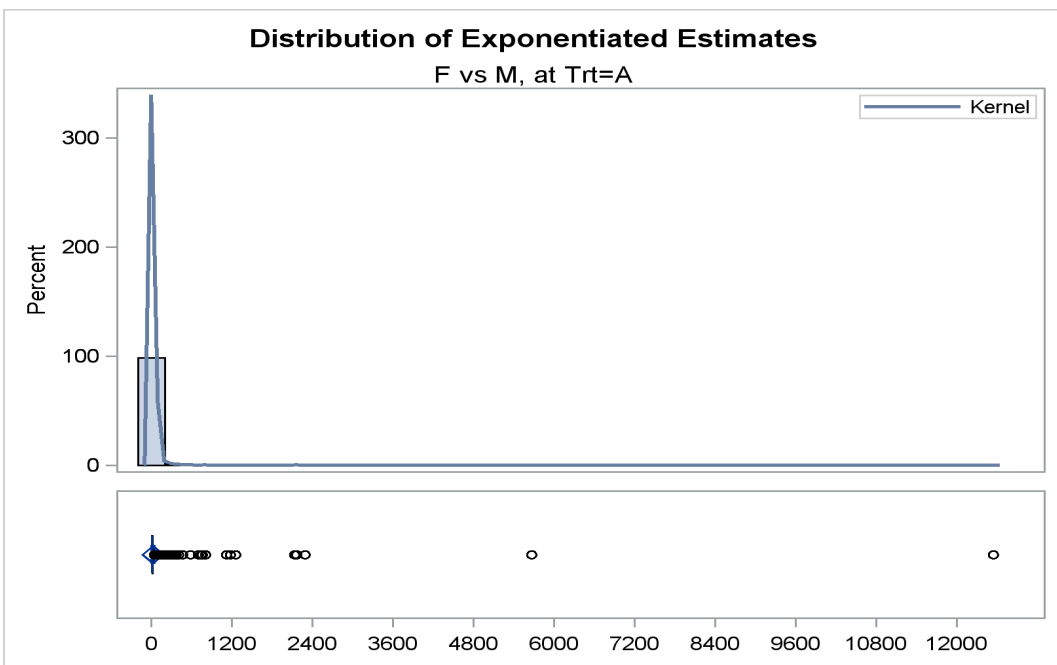
Odds Ratio

Female vs. Male, at Treatment = A.

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

Sample Estimate							
			Percentiles for Exponentiated				
Label	Exponentiated	Standard Deviation of Exponentiated	25th	50th	75th	Lower HPD of Exponentiated	Upper HPD of Exponentiated
F vs M, at Trt=A	28.3873	188.824003	2.1744	5.9664	18.4925	0.1876	93.1034

Histogram of the Posterior Odds Ratio



Odds Ratio

Similarly, you can estimate odds ratios conditional on different treatements:

```
proc plm restore=logit_bayes;
   estimate "F vs M, at Trt=B"
             sex 1 -1 treatment*sex [1, 2 1] [-1, 2 2] /exp;
   estimate "F vs M, at Trt=P"
             sex 1 -1 treatment*sex [1, 3 1] [-1, 3 2] /exp;
   run;
```


Odds Ratio

Female vs. Male, at Treatment = B.

Sample Estimate									
				Percentiles					
Label	N	Estimate	Standard Deviation	25th	50th	75th	Alpha	Lower HPD	Upper HPD
F vs M, at Trt=B	10000	2.5670	1.5778	1.4946	2.4569	3.5345	0.05	-0.1317	5.9040

Sample Estimate							
			Percentiles for Exponentiated				
Label	Exponentiated	Standard Deviation of Exponentiated	25th	50th	75th	Lower HPD of Exponentiated	Upper HPD of Exponentiated
F vs M, at Trt=B	60.4417	384.724355	4.4575	11.6684	34.2779	0.1399	195.64

Odds Ratio

Female vs. Male, at Treatment = P.

Sample Estimate									
				Percentiles					
Label	N	Estimate	Standard Deviation	25th	50th	75th	Alpha	Lower HPD	Upper HPD
F vs M, at Trt=P	10000	2.9148	1.7137	1.7348	2.8056	3.9222	0.05	-0.1694	6.5110

Sample Estimate							
			Percentiles for Exponentiated				
Label	Exponentiated	Standard Deviation of Exponentiated	25th	50th	75th	Lower HPD of Exponentiated	Upper HPD of Exponentiated
F vs M, at Trt=P	175.97	1642.867153	5.6676	16.5362	50.5135	0.3686	408.55

Outline

Cox Model

Consider the data for the Veterans Administration lung cancer trial presented in Appendix 1 of Kalbfleisch and Prentice (1980).

Time	Death in days
Therapy	Type of therapy: standard or test
Cell	Type of tumor cell: adeno, large, small, or squamous
PTherapy	Prior therapy: yes or no
Age	Age in years
Duration	Months from diagnosis to randomization
KPS	Karnofsky performance scale
Status	Censoring indicator (1=censored time, 0=event time)

Cox Model

A subset of the 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
...								

- Some parameters are the coefficients of the continuous variables (`KPS`, `Duration`, and `Age`).
- Other parameters are the coefficients of the design variables for the categorical explanatory variables (`PTherapy`, `Cell`, and `Therapy`).

Cox Model

The model considered here is the Breslow partial likelihood:

$$L(\beta) = \prod_{i=1}^{k} \frac{e^{\beta' \sum_{j \in \mathcal{D}_i} \mathbf{Z}_j(t_i)}}{\left[\sum_{l \in \mathcal{R}_i} e^{\beta' \mathbf{Z}_l(t_i)} \right]^{d_i}}$$

where

- $t_1 < \cdots < t_k$ are distinct event times
- $\mathbf{Z}_j(t_i)$ is the vector explanatory variables for the jth individual at time t_i
- $\mathcal{R}_i$ is the risk set at t_i, which includes all observations that have survival time greater than or equal to t_i
- d_i is the multiplicity of failures at t_i. It is the size of the set $\mathcal{D}_i$ of individuals that fail at t_i

Cox Model

The following statements fit a Cox regression model with a uniform prior on the regression coefficients:

```
proc phreg data=VALung;
   class PTherapy(ref='no') Cell(ref='large')
         Therapy(ref='standard');
   model Time*Status(0) = KPS Duration Age PTherapy Cell Therapy;
   bayes seed=1 outpost=cout coeffprior=uniform;
run;
```

Cox Model: Posterior Mean Estimates

Bayesian Analysis

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
Kps	10000	-0.0327	0.00545	-0.0364	-0.0328	-0.0291
Duration	10000	-0.00170	0.00945	-0.00791	-0.00123	0.00489
Age	10000	-0.00852	0.00935	-0.0147	-0.00850	-0.00223
Ptherapyyes	10000	0.0754	0.2345	-0.0776	0.0766	0.2340
Celladeno	10000	0.7867	0.3080	0.5764	0.7815	0.9940
Cellsmall	10000	0.4632	0.2731	0.2775	0.4602	0.6435
Cellsquamous	10000	-0.4022	0.2843	-0.5935	-0.4024	-0.2124
Therapytest	10000	0.2897	0.2091	0.1500	0.2900	0.4294

Cox Model: Interval Estimates

Bayesian Analysis

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Kps	0.050	-0.0433	-0.0219	-0.0434	-0.0221
Duration	0.050	-0.0216	0.0153	-0.0202	0.0164
Age	0.050	-0.0271	0.00980	-0.0270	0.00983
Ptherapyyes	0.050	-0.3943	0.5335	-0.3715	0.5488
Celladeno	0.050	0.1905	1.3969	0.1579	1.3587
Cellsmall	0.050	-0.0617	1.0039	-0.0530	1.0118
Cellsquamous	0.050	-0.9651	0.1519	-0.9550	0.1582
Therapytest	0.050	-0.1191	0.6955	-0.1144	0.6987

Cox Model: Plotting Survival Curves

Suppose that you are interested in estimating the survival curves for two individuals who have similar characteristics, with one receiving the standard treatment while the other did not. The following is saved in the SAS data set pred:

OBS	Ptherapy	kps	duration	age	cell	therapy
1	no	58	8.7	60	large	standard
2	no	58	8.7	60	large	test

Cox Model

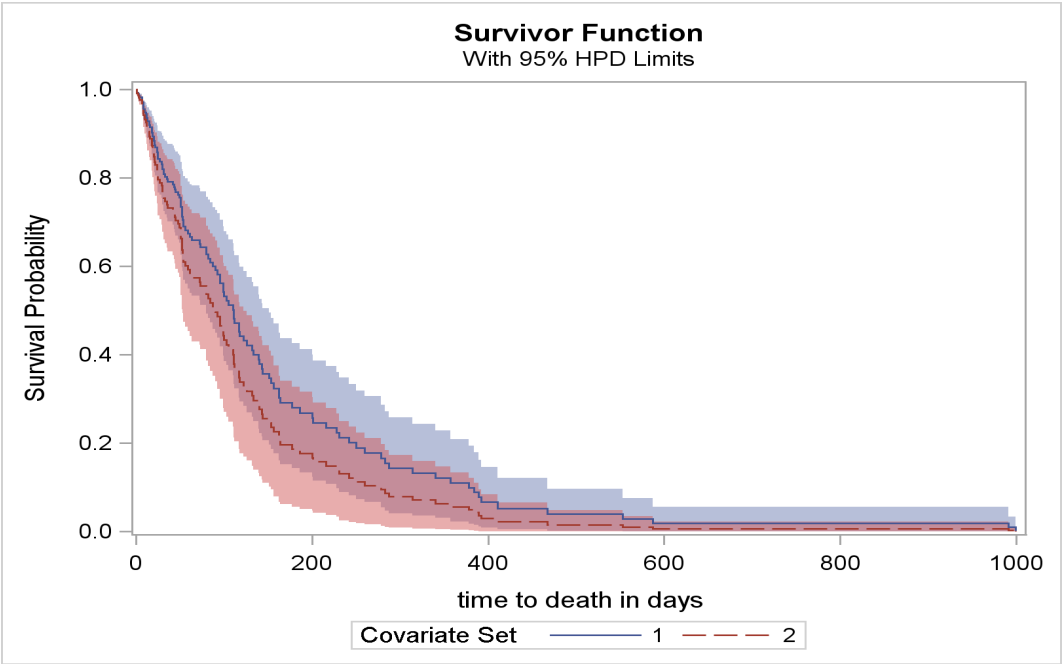
You can use the following statements to estimate the survival curves and save the estimates to a SAS data set:

```
proc phreg data=VALung plots(cl=hpd overlay)=survival;
   baseline covariates=pred out=pout;
   class PTherapy(ref='no') Cell(ref='large')
         Therapy(ref='standard');
   model Time*Status(0) = KPS Duration Age PTherapy Cell Therapy;
   bayes seed=1 outpost=cout coeffprior=uniform;
run;
```

plots : requests survival curves with overlaying HPD intervals

baseline : specifies input covariates data set and saves the posterior prediction to the OUT= data set

Cox Model: Posterior Survival Curves



Estimated survival curves for the two subjects and their corresponding 95% HPD intervals.

Hazard Ratios

The HAZARDRATIO statement enables you to obtain customized hazard ratios, ratios of two hazard functions.

HAZARDRATIO <'label'> variables < / options > ;

- For a continuous variable: the hazard ratio compares the hazards for a given change (by default, a increase of 1 unit) in the variable.
- For a CLASS variable, a hazard ratio compares the hazards of two levels of the variable.

Hazard Ratios

The following SAS statements fit the same Cox regression model and request three kinds of hazard ratios.

```
proc phreg data=VALung;
   class PTherapy(ref='no') Cell(ref='large')
         Therapy(ref='standard');
   model Time*Status(0) = KPS Duration Age PTherapy Cell Therapy;
   bayes seed=1 outpost=vout plots=trace coeffprior=uniform;
   hazardratio 'HR 1' Therapy / at(PTherapy='yes' KPS=80
                            duration=12 age=65 cell='small');
   hazardratio 'HR 2' Age / unit=10 at(KPS=45);
   hazardratio 'HR 3' Cell;
run;
```


Hazard Ratios

The following results are the summary statistics of the posterior hazards between the standard therapy and the test therapy.

Bayesian Analysis

HR 1: Hazard Ratios for Therapy

Description	N	Mean	Standard Deviation	Quantiles 25%	50%	75%
Therapy standard vs test At Prior=yes Kps=80 Duration=12 Age=65 Cell=small	10000	0.7651	0.1617	0.6509	0.7483	0.8607

HR 1: Hazard Ratios for Therapy

95% Equal-Tail Interval		95% HPD Interval	
0.4988	1.1265	0.4692	1.0859

Hazard Ratios

The following table lists the change of hazards for an increase in Age of 10 years.

Bayesian Analysis

HR 2: Hazard Ratios for Age										
				Quantiles						
Description	N	Mean	Standard Deviation	25%	50%	75%	95% Equal-Tail Interval		95% HPD Interval	
Age Unit=10 At Kps=45	10000	0.9224	0.0865	0.8633	0.9185	0.9779	0.7629	1.1030	0.7539	1.0904

Hazard Ratios

The following table lists posterior hazards between different levels in the Cell variable:

Bayesian Analysis

HR 3: Hazard Ratios for Cell										
				Quantiles						
Description	N	Mean	Standard Deviation	25%	50%	75%	95% Equal-Tail Interval		95% HPD Interval	
Cell adeno vs large	10000	2.3035	0.7355	1.7797	2.1848	2.7020	1.2099	4.0428	1.0661	3.7509
Cell adeno vs small	10000	1.4374	0.4124	1.1479	1.3811	1.6622	0.7985	2.3857	0.7047	2.2312
Cell adeno vs squamous	10000	3.4376	1.0682	2.6679	3.2903	4.0199	1.8150	5.9733	1.6274	5.6019
Cell large vs small	10000	0.6530	0.1798	0.5254	0.6311	0.7577	0.3664	1.0636	0.3357	1.0141
Cell large vs squamous	10000	1.5567	0.4514	1.2367	1.4954	1.8103	0.8591	2.6251	0.7776	2.4679
Cell small vs squamous	10000	2.4696	0.7046	1.9717	2.3742	2.8492	1.3872	4.1403	1.2958	3.9351

Outline

Piecewise Exponential Model (Optional)

Let $\{(t_i, \mathbf{x}_i, \delta_i), i = 1, 2, \ldots, n\}$ be the observed data. Let $a_0 = 0 < a_1 < \ldots < a_{J-1} < a_J = \infty$ be a partition of the time axis. The hazard for subject i is

$$h(t|\mathbf{x}_i; \theta) = h_0(t) \exp(\beta' \mathbf{x}_i)$$

where

$$h_0(t) = \lambda_j \qquad a_{j-1} \leq t < a_j \qquad (j = 1, \ldots, J)$$

The hazard for subject i in the jth time interval is

$$h(t) = \lambda_j \exp(\beta' \mathbf{x}_i) \qquad a_{j-1} < t < a_j$$

Piecewise Exponential Model

From the hazard function, first define the baseline cumulative hazard function:

$$H_0(t) = \sum_{j=1}^{J} \lambda_j \Delta_j(t)$$

where

$$\Delta_j(t) = \begin{cases} 0 & t < a_{j-1} \\ t - a_{j-1} & a_{j-1} \leq t < a_j \\ a_j - a_{j-1} & t \geq a_j \end{cases}$$

Piecewise Exponential Model

The log likelihood is:

$$\begin{aligned} l(\boldsymbol{\lambda}, \boldsymbol{\beta}) &= \sum_{i=1}^{n} \delta_i \left[\sum_{j=1}^{J} I(a_{j-1} \leq t_i < a_j) \log \lambda_j + \boldsymbol{\beta}' \mathbf{x}_i \right] \\ &\quad - \sum_{i=1}^{n} \left[\sum_{j=1}^{J} \Delta_j(t_i) \lambda_j \right] \exp(\boldsymbol{\beta}' \mathbf{x}_i) \end{aligned}$$

where δ_i is the event status:

$$\delta_i = \begin{cases} 0 & \text{if } t_i \text{ is a censored time} \\ 1 & \text{if } t_i \text{ is an event time} \end{cases}$$

This model has two parameter vectors: $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$.

Piecewise Exponential Model

PROC PHREG supports the following priors for the piecewise exponential model:

- Regression coefficients ($\boldsymbol{\beta}$): normal and uniform priors
- Hazards ($\boldsymbol{\lambda}$): improper, uniform, independent gamma, and AR(1) priors
- Log hazards ($\boldsymbol{\alpha} = \log(\boldsymbol{\lambda})$): uniform and normal priors
- Regression coefficients and log hazards: multivariate normal (do not need to be independent)

Piecewise Exponential Model

Consider a randomized trial of 40 rats exposed to carcinogen:

- Drug X and Placebo are the treatment groups.
- Event of interest is death.
- Response is time until death.
- What are the effects of treatment and gender on survival?

Piecewise Exponential Model

A subset of the data:

```
proc format;
   value Rx 1='X' 0='Placebo';
data Exposed;
  input Days  Status Trt Gender $ @@;
  format Trt Rx.;
  datalines;
179   1   1   F   378   0   1   M
256   1   1   F   355   1   1   M
262   1   1   M   319   1   1   M
256   1   1   F   256   1   1   M
...
268   0   0   M   209   1   0   F
;
```


Piecewise Exponential Model

An appropriate model is the piecewise exponential. In the model:

- Each time interval has a constant hazard
- There are a total of eight intervals (PROC PHREG default)
- Intervals are determined by placing roughly equal number of uncensored observations in each interval
- The log hazard is used. It is generally more computationally stable. There are 8 λ_i's and two regression coefficients.

Piecewise Exponential Model

The following programming statements fit a Bayesian piecewise exponential model with noninformative priors on both β and $\log(\lambda)$:

```
proc phreg data=Exposed;
   class Trt(ref='Placebo') Gender(ref='F');
   model Days*Status(0)=Trt Gender;
   bayes seed=1 outpost=eout piecewise=loghazard(n=8);
run;
```

The PIECEWISE= option requests the estimating of a piecewise exponential model with 8 intervals.

Piecewise Exponential Model

Suppose that you have some prior information w.r.t. both β and $\log(\lambda)$ that can be approximated well with a multivariate normal distribution. You can construct the following data set:

```
data pinfo;
   input _TYPE_ $ alpha1-alpha8 trtX GenderM;
   datalines;
   Mean 0 0 0 0 0 0 0 0 0 0
   cov  90.2  -9.8   1.3  -1.9   4.1   3.7  14.3 -10.7  -7.2  -4.2
   cov  -9.8 102.4  15.3 -12.1  15.6   6.8 -23.7 -23.7   9.0  -8.8
   cov   1.3  15.3 102.8  13.0  22.1   5.7  21.4 -16.1  14.2  13.3
   cov  -1.9 -12.1  13.0  90.2   4.6 -16.1  11.3  -8.6 -12.6  -1.2
   cov   4.1  15.6  22.1   4.6 107.9  18.2   2.4  -8.1   2.9 -16.4
   cov   3.7   6.8   5.7 -16.1  18.2 123.3  -2.7  -7.9   3.2  -3.4
   cov  14.3 -23.7  21.4  11.3   2.4  -2.7 114.2   2.3   6.7  11.6
   cov -10.7 -23.7 -16.1  -8.6  -8.1  -7.9   2.3  91.8  -7.6   0.0
   cov  -7.2   9.0  14.2 -12.6   2.9   3.2   6.7  -7.6 100.0  -6.3
   cov  -4.2  -8.8  13.3  -1.2 -16.4  -3.4  11.6   0.0  -6.3 124.7
   ;
```

Piecewise Exponential Model

The following programming statements fit a Bayesian piecewise exponential model with informative prior on both β and $\log(\lambda)$:

```
proc phreg data=exposed;
   class Trt(ref='Placebo') Gender(ref='F');
   model Days*Status(0)=Trt Gender;
   bayes seed=1 outpost=eout
      piecewise=loghazard(n=8 prior=normal(input=pinfo))
      cprior=normal(input=pinfo);
run;
```


Piecewise Exponential Model

Bayesian Analysis

Model Information	
Data Set	WORK.EXPOSED
Dependent Variable	Days
Censoring Variable	Status
Censoring Value(s)	0
Model	Piecewise Exponential
Burn-In Size	2000
MC Sample Size	10000
Thinning	1

Summary of the Number of Event and Censored Values			
Total	Event	Censored	Percent Censored
40	36	4	10.00

Piecewise Exponential Model

The partition of the time intervals:

Bayesian Analysis

Constant Hazard Time Intervals				
Interval				
[Lower,	Upper)	N	Event	Log Hazard Parameter
0	193	5	5	Alpha1
193	221	5	5	Alpha2
221	239.5	7	5	Alpha3
239.5	255.5	5	5	Alpha4
255.5	256.5	4	4	Alpha5
256.5	278.5	5	4	Alpha6
278.5	321	4	4	Alpha7
321	Infty	5	4	Alpha8

Piecewise Exponential Model

Posterior summary statistics:

Bayesian Analysis

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
Alpha1	10000	-6.4137	0.4750	-6.7077	-6.3770	-6.0852
Alpha2	10000	-4.0505	0.4870	-4.3592	-4.0207	-3.7058
Alpha3	10000	-2.9297	0.5146	-3.2468	-2.8954	-2.5737
Alpha4	10000	-1.9146	0.6212	-2.3256	-1.8936	-1.4839
Alpha5	10000	1.2433	0.6977	0.7948	1.2598	1.7255
Alpha6	10000	-0.8729	0.8040	-1.4033	-0.8692	-0.3276
Alpha7	10000	-0.9827	0.8346	-1.5247	-0.9646	-0.4223
Alpha8	10000	0.4771	0.9095	-0.1262	0.4796	1.0952
TrtX	10000	-1.2319	0.3929	-1.4898	-1.2286	-0.9707
GenderM	10000	-2.6607	0.5483	-3.0159	-2.6466	-2.2888

Piecewise Exponential Model

Interval estimates:

Bayesian Analysis

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Alpha1	0.050	-7.4529	-5.5710	-7.3576	-5.5100
Alpha2	0.050	-5.0961	-3.1973	-5.0030	-3.1303
Alpha3	0.050	-4.0327	-2.0130	-3.9950	-1.9843
Alpha4	0.050	-3.1799	-0.7614	-3.1671	-0.7536
Alpha5	0.050	-0.1893	2.5585	-0.0872	2.6410
Alpha6	0.050	-2.4616	0.6875	-2.4942	0.6462
Alpha7	0.050	-2.6588	0.6248	-2.6383	0.6400
Alpha8	0.050	-1.3264	2.2243	-1.2867	2.2359
TrtX	0.050	-2.0147	-0.4735	-2.0195	-0.4849
GenderM	0.050	-3.7758	-1.6150	-3.7269	-1.5774

Piecewise Exponential Model

Hazard ratios of Treatment and Gender:

```
hazardratio 'Hazard Ratio Statement 1' Trt;
hazardratio 'Hazard Ratio Statement 2' Gender;
```

Bayesian Analysis

Hazard Ratio Statement 1: Hazard Ratios for Trt										
				Quantiles						
Description	N	Mean	Standard Deviation	25%	50%	75%	95% Equal-Tail Interval		95% HPD Interval	
Trt Placebo vs X	10000	3.7058	1.5430	2.6399	3.4164	4.4362	1.6056	7.4981	1.3129	6.7830

Hazard Ratio Statement 2: Hazard Ratios for Gender										
				Quantiles						
Description	N	Mean	Standard Deviation	25%	50%	75%	95% Equal-Tail Interval		95% HPD Interval	
Gender F vs M	10000	16.6966	10.3427	9.8629	14.1062	20.4071	5.0281	43.6302	3.4855	36.3649

Outline

The MCMC Procedure

The MCMC procedure (SAS/STAT®9.2, 9.22, 9.3, 12.1) is a simulation procedure that can be used to fit:

- single-level or multilevel (hierarchical) models
- linear or nonlinear models, such as regression, survival, ordinal multinomial, and so on.
- missing data problems

The procedure selects appropriate sampling algorithms for the models that you specified, and it is capable of executing SAS DATA step language for estimation and inference.

PROC MCMC Statements

PROC MCMC *options*;

PARMS; define parameters.

PRIOR; declare prior distributions

Programming statements;
MODEL
} define log-likelihood function

PREDDIST; posterior prediction

RANDOM; random effects

Run;

Linear Regression

$$
\begin{aligned}
\text{weight}_i &\sim \text{normal}(\mu_i, \text{var} = \sigma^2) \\
\mu &= \beta \cdot \text{height}_i \\
\beta &\sim \text{normal}(0, \text{var} = 100) \\
\sigma^2 &\sim \text{inverse Gamma}(\text{shape} = 2, \text{scale} = 2)
\end{aligned}
$$

The data:

```
data class;
   input height weight;
datalines;
   69.0      112.5
   56.5       84.0
...
   66.5      112.0
;
```

MCMC Program:

```
proc mcmc data=class seed=1 nbi=5000
    nmc=10000 outpost=regOut;
   parms beta s2;
   prior beta ~ normal(0, var=100);
   prior s2 ~ igamma(shape=2, scale=2);
   mu = beta * height;
   model weight ~ normal(mu, var=s2);
   run;
```

Linear Regression

$$\begin{aligned} \text{weight}_i &\sim \text{t}(\mu_i, \text{sd} = \sigma, \text{df} = 3) \\ \mu &= \beta \cdot \text{height}_i \\ \beta &\sim \text{normal}(0, \text{var} = 100) \\ \sigma &\sim \text{uniform}(0, 25) \end{aligned}$$

Change the model, parameterization, and so on as you please:

```
proc mcmc data=class seed=1 nbi=5000 nmc=10000 outpost=regOut;
   parms beta sig;
   prior beta ~ normal(0, var=100);
   prior sig ~ uniform(0, 25);
   mu = beta * height;
   model weight ~ t(mu, sd=sig, df=3);
   run;
```


The Posterior Distribution

PROC MCMC is sampling-based procedure, which is similar to other SAS Bayesian procedures. BUT, you must be more aware of how the posterior distribution is constructed:

$$\pi(\theta|\mathbf{y}, \mathbf{x}) \propto \pi(\theta) \cdot f(\mathbf{y}|\theta, \mathbf{x})$$

- The PRIOR statements define the prior distributions: $\pi(\theta)$.
- The MODEL statement defines the likelihood function for each observation in the data set: $f(y_i|\theta, x_i)$, for $i = 1, \cdots, n$
- The procedure calculates the posterior distribution (on the log scale):

$$\log(\pi(\theta|\mathbf{y}, \mathbf{x})) = \log(\pi(\theta)) + \sum_{i=1}^{n} \log(f(y_i|\theta, x_i))$$

where $\mathbf{y} = \{y_i\}$ and $\mathbf{x} = \{x_i\}$

Calculate of $\log(\pi(\theta|\mathbf{y}))$

At each iteration, the programming and MODEL statements are executed for each observation to obtain $\log(\pi(\theta|\mathbf{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0

```
proc mcmc data=input;
   prior;
   progm stmt;
   model ;
   run;
```

at the top of the data set

$\log \pi(\theta|\mathbf{y}) = \log(f(y_1|\theta))$

Calculate of $\log(\pi(\theta|\mathbf{y}))$

At each iteration, the programming and MODEL statements are executed for each observation to obtain $\log(\pi(\theta|\mathbf{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0

```
proc mcmc data=input;
   prior;
   progm stmt;
   model ;
   run;
```

stepping through the data set

$\log \pi(\theta|\mathbf{y}) = \log \pi(\theta|\mathbf{y}) + \log(f(y_2|\theta))$

Calculate of $\log(\pi(\theta|\mathbf{y}))$

At each iteration, the programming and MODEL statements are executed for each observation to obtain $\log(\pi(\theta|\mathbf{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0

```
proc mcmc data=input;
   prior;
   progm stmt;
   model ;
   run;
```

stepping through the data set

$\log \pi(\theta|\mathbf{y}) = \log \pi(\theta|\mathbf{y}) + \log(f(y_3|\theta))$

Calculate of $\log(\pi(\theta|\mathbf{y}))$

At each iteration, the programming and MODEL statements are executed for each observation to obtain $\log(\pi(\theta|\mathbf{y}))$

Obs	Height	Weight
1	69.0	112.5
2	56.5	84.0
3	65.3	98.0
...		
19	66.5	112.0

```
proc mcmc data=input;
   prior;
   progm stmt;
   model;
   run;
```

at the last observation, the prior is included

$\log \pi(\theta|\mathbf{y}) = \log(\pi(\theta)) + \sum_{i=1}^{n} \log(f(y_i|\theta))$

PROC MCMC and WinBUGS Syntax are Similar

Both require going through the data set (repeatedly). In WinBUGS, a for-loop and array indices are used to access records in variables; In PROC MCMC, the looping over the data set is hidden behind the scene.

```
height[] weight[]
  69.0      112.5
  56.5       84.0
  65.3       98.0
...
  66.5      112.0
END
```

```
model
{
   for(i in 1:19) {
      mu[i] = beta * height[i]
      weight[i] ~ dnorm(mu[i], tau)
   }
   beta ~ dnorm(0, 0.1)
   tau ~ gamma(0.1, 0.1)
}
```


Sampling in PROC MCMC

PROC MCMC recognizes certain configurations of the statistical models and applies sampling methods (conjugate or direct) when appropriate.

In other cases, the default sampling algorithm a normal-kernel-based random walk Metropolis. The proposal distribution is $q(\theta_{\text{new}}|\theta^{(t)}) = \text{MVN}(\theta_{\text{new}}|\theta^{(t)}, c^2\mathbf{\Sigma})$.

Two components in the Metropolis algorithm:

- construction of the proposal distribution—automatically done by PROC MCMC
- evaluation of $\log(\pi(\theta^{(t)}|\mathbf{y}))$ at each iteration—specified via the PRIOR and MODEL statements

PARMS Statement

PARMS *name* | *(name-list)* <=> *number*;

- lists the names of the parameters
- specifies optional initial values
- specifies updating sequence of the parameters

For example:

```
PARMS alpha 0 beta 1;
```

declares α and β to be model parameters and assigns 0 to α and 1 to β.

```
PARMS alpha 0 beta;
```

assigns 0 to α and leaves β uninitialized.

```
PARMS (alpha beta) 1;
```

assigns 1 to both α and β.

PARMS Statement

When multiple PARMS statements are used, each statement defines a block of parameters, which are updated sequentially in each iteration:

```
PARMS beta0 beta1;
PARMS sigma2;
```

At each iteration t, PROC MCMC updates β_0 and β_1 together, alternatively with σ^2, each with a Metropolis sampler:

$$\begin{aligned} \beta_0^{(t)}, \beta_1^{(t)} &\mid \sigma^2_{(t-1)}, \text{Data} \\ \sigma^2_{(t)} &\mid \beta_0^{(t)}, \beta_1^{(t)}, \text{Data} \end{aligned}$$

PRIOR Statement

PRIOR *parameter-list* $\sim$ *distribution*;

specifies the prior distributions of model parameters. For example:

```
PRIOR alpha ~ normal(0, var=10);
PRIOR sigma2 ~ igamma(0.001, iscale=0.001);
PRIOR beta gamma ~ normal(alpha, var=sigma2);
```

specifies the following joint prior distribution:

$$\pi(\alpha, \beta, \gamma, \sigma^2) = \pi(\beta|\alpha, \sigma^2) \cdot \pi(\gamma|\alpha, \sigma^2) \cdot \pi(\alpha) \cdot \pi(\sigma^2)$$

MODEL Statement

MODEL *dependent-variable-list* $\sim$ *distribution*;

specifies the likelihood function. The dependent variables can be

- data set variables

```
MODEL y ~ normal(alpha, var=1);
```

- functions of data set variables

```
w = log(y);
MODEL w ~ normal(alpha, var=1);
```

You can specify multiple MODEL statements.

Standard Distributions

Standard distributions in the PRIOR and MODEL statements:

beta	binary	binomial	cauchy	chisq
expon	gamma	geo	ichisq	igamma
laplace	negbin	normal	pareto	poisson
sichisq	t	uniform	wald	weibull
dirich	iwish	mvn	mvnar	multinom[2]

Distribution argument can be constants, expressions, or model parameters. For example:

```
prior alpha ~ cauchy(0, 2);
prior p ~ beta(abs(alpha), constant('pi'));
model y ~ binomial(n, p);
```

[2] Only in the MODEL statement

Standard Distributions

Some distributions can be parameterized in different ways:

expon(scale|s $= \lambda$)	expon(iscale|is $= \lambda$)	
gamma(a, scale|sc $= \lambda$)	gamma(a, iscale|is $= \lambda$)	
igamma(a, scale|sc $= \lambda$)	igamma(a, iscale|is $= \lambda$)	
laplace(l, scale|sc $= \lambda$)	laplace(l, iscale|is $= \lambda$)	
normal(μ, var=σ^2)	normal(μ, sd=σ)	normal(μ, prec=τ)
lognormal(μ, var=σ^2)	lognormal(μ, sd=σ)	lognormal(μ, prec=τ)
t(μ, var=σ^2, df)	t(μ, sd=σ, df)	t(μ, prec=τ, df)

For these distributions, you must explicitly name the ambiguous parameter. For example:

```
prior beta ~ normal(0, var=sigma2);
prior sigma2 ~ igamma(0.001, is=0.001);
```

Truncated Distributions

Univariate distributions allow for optional LOWER= and UPPER= arguments.

```
prior p ~ beta(2,3, lower=0.5);
prior b ~ expon(scale=100, lower=100, upper=2000);
```

The bounds can be random variables (parameters):

```
prior alpha ~ normal(0, sd=1);
prior beta ~ normal(0, sd=1, lower=alpha);
```


Programming Statements

Most DATA step operators, functions, and statements can be used in PROC MCMC:

- assignment and operators: +, -, *, /, <>, <, ...
- mathematical functions: ABS, LOG, PDF, CDF, SDF, LOGPDF, ...
- statements: CALL, DO, IF, PUT, WHEN, ...

The functions enable you to:

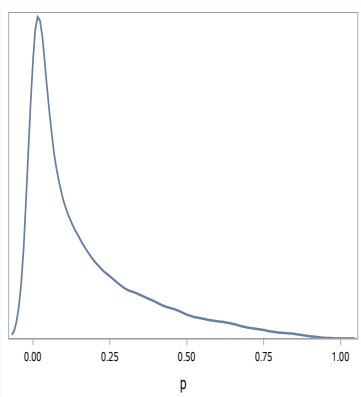
- compute functions of parameters
- construct general prior and/or likelihood functions

Outline

Monte Carlo Simulation

$$p \sim \text{beta}(0.47, 2.35)$$

```
data a;
   run;
proc mcmc data=a seed=17 nmc=20000;
   parm p;
   prior p ~ beta(0.47, 2.35);
   model general(0);
run;
```



Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
p	20000	0.1662	0.1912	0.0199	0.0906	0.2512

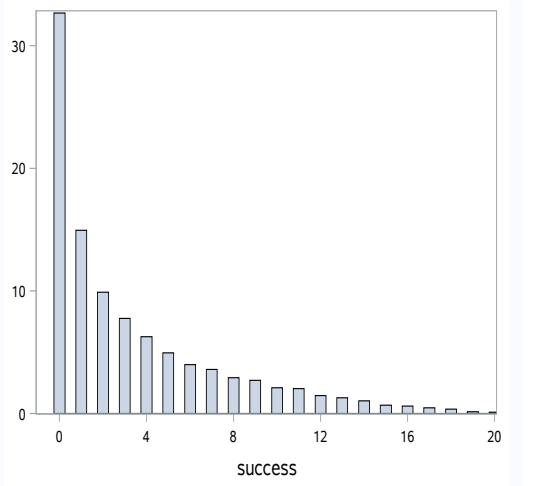
Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
p	0.050	0.000152	0.6866	1.38E-10	0.5890

Simple Simulation

If p represents some process of binary success probabilities, then you might be interested in the outcome of a binomial trial where the probability is not known precisely:

$$p \sim \text{beta}(0.47, 2.35)$$
$$success \sim \text{binomial}(20, p)$$

```
proc mcmc data=a seed=17 nmc=20000
   outpost=o1;
   parm p success;
   prior p ~ beta(0.47, 2.35);
   prior success ~ binomial(20, p);
   model general(0);
run;
```



Estimates of p and *success*

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
p	20000	0.1676	0.1908	0.0208	0.0942	0.2532
success	20000	3.3545	4.0982	0	2.0000	5.0000

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
p	0.050	0.000157	0.6832	1.38E-10	0.5834
success	0.050	0	14.0000	0	12.0000

Estimate Cumulative Probability

$$\begin{aligned} p &\sim \text{beta}(0.47, 2.35) \\ \textit{success} &\sim \text{binomial}(20, p) \end{aligned}$$

what is the $\Pr(9 \leq \textit{success} \leq 12)$?

```
proc mcmc data=a seed=17 nmc=20000 outpost=o1 monitor=(prob);
   parm p success;
   prior p ~ beta(0.47, 2.35);
   prior success ~ binomial(20, p);
   prob = (9 <= success <= 12);
   model general(0);
run;
```

monitor keeps track of variables in a program

The estimated probability is 0.083.

First few samples of the OUTPOST data set:

Obs	Iteration	p	success	prob
1	1	0.00691	0	0
2	2	0.00369	0	0
3	3	0.4078	8.0000	0
4	4	0.0280	1.0000	0
5	5	0.0875	2.0000	0
...				

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
prob	20000	0.0830	0.2759	0	0	0

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
prob	0.050	0	1.0000	0	1.0000

Outline

Binomial model

Researchers are interested in evaluating the performance of a medical procedure in a multicenter study. The following statements create a data set for the treatment arm of the trials:

```
data trials;
   input event n center;
   datalines;
   2  86  1
   2  69  2
   1  71  3
   1 113  4
   1 103  5
;
```

event: number of deaths

n: number of patients assigned to the treatment procedure

center: center index

Binomial Example

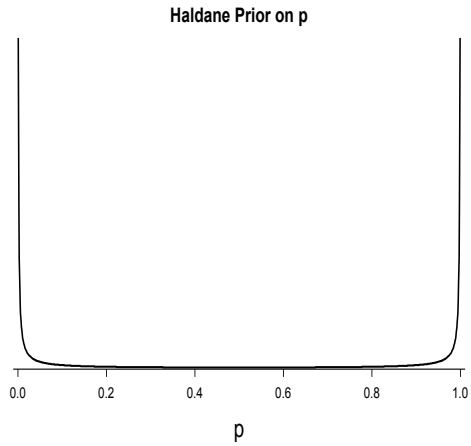
Consider a simple binomial model

$$\begin{aligned} \text{event}_i &\sim \text{binomial}(n_i, p) \\ p &\sim \text{beta}(a, b) \end{aligned}$$

where p is the parameter of interest and a and b are hyper-parameters. Consider the following choices for a and b:

- uniform: beta$(1, 1)$
- Haldane prior: beta$(0, 0)$

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



Binomial Model with Flat Prior

```
proc mcmc data=trials seed=17 nmc=20000 outpost=UnifBin;
   parm p;
   prior p ~ beta(1,1);
   model event ~ binomial(n,p);
   run;
```

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
p	20000	0.0180	0.00624	0.0135	0.0174	0.0217

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
p	0.050	0.00784	0.0321	0.00670	0.0303

Binomial Model with Haldane Prior

The following syntax does not work in PROC MCMC,

```
prior p ~ beta(0, 0);
```

because the shape and scale parameters of a beta distribution must be positive.

Use the GENERAL function to construct nonstandard prior distribution.

Specifying a Nonstandard Distribution

The GENERAL and DGENERAL functions enable you construct your own prior or likelihood function. The "D" stands for discrete.

PRIOR `alpha` $\sim$ `dgeneral(lp);`
MODEL `y` $\sim$ `general(llike);`

The expressions `lp` and `llike` must take the values of the **logarithm** of the distribution.

The normalizing constant of the distribution can be ignored, as long as it is independent of other parameters in the model.

The GENERAL Distribution

Suppose that you want to use the following prior:

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2}$$

which is a nonstandard distribution (nonintegrable prior). The logarithm of this prior is

$$\log(\pi(\sigma^2)) = -\log(\sigma^2) + C$$

You use the following statements to declare this prior:

```
lp = -log(sigma2);
prior sigma2 ~ general(lp, lower=0);
```


More on the GENERAL Distribution

The function argument can be an expression or a constant. For example, to specify $\pi(\alpha) \propto 1$, you use the following statement:

```
prior alpha ~ general(0);
```

Use these functions with care because PROC MCMC cannot verify that the priors you specify lead to valid (integrable) posterior.

When in doubt, stay with proper distributions.

Binomial Model with Haldane Prior

$$\pi(p) \propto p^{-1}(1-p)^{-1}$$
$$\Rightarrow \quad \log(\pi(p)) = -(log(p) + log(1-p))$$

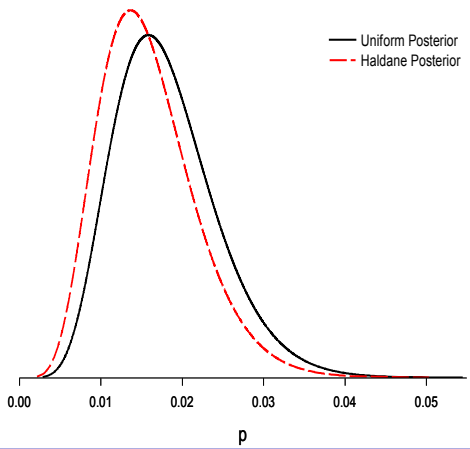
```
proc mcmc data=trials seed=17 nmc=20000 outpost=HalBin;
   parm p 0.5;
   lprior = -(log(p) + log(1-p));
   prior p ~ general(lprior, lower=0, upper=1);
   model event ~ binomial(n,p);
   run;
```

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
p	20000	0.0158	0.00595	0.0114	0.0150	0.0193

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
p	0.050	0.00640	0.0294	0.00579	0.0280



Binomial Model

Suppose that you do not want to have fixed hyperparameter values and want to consider hyperprior distributions on these parameters:

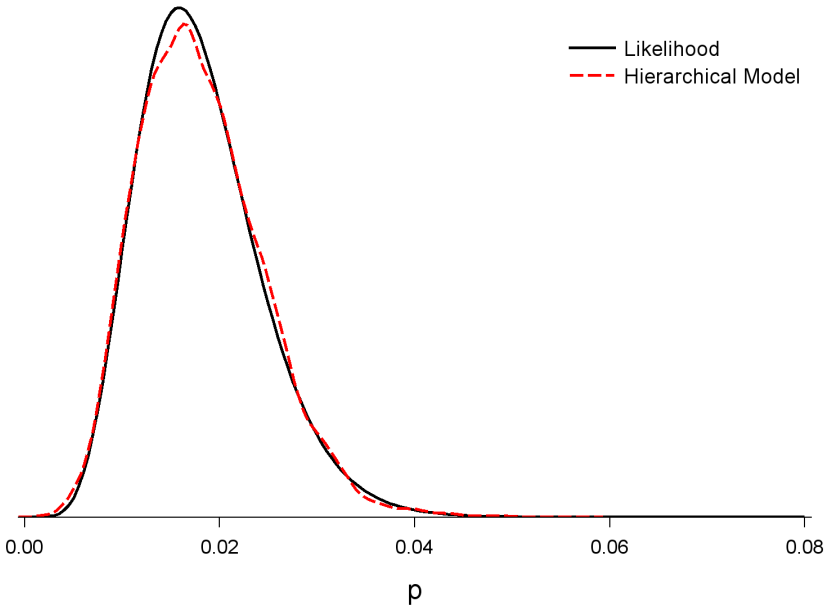
$$\pi(a) \propto \text{exponential}(\text{scale} = 100)$$
$$\pi(b) \propto \text{exponential}(\text{scale} = 100)$$

This prior has mean of 100 and variance 10,000. The following statements fit a hierarchical binomial model:

```
proc mcmc data=trials seed=17 nmc=10000 outpost=bmc;
   parms p;
   parms a b;
   prior a b ~ expon(scale=100);
   prior p ~ beta(a,b);
   model event ~ binomial(n,p);
   run;
```

Posterior Density Comparison

Having hyperprior distributions is essentially equivalent to using a uniform prior on p—there is no information in the data that can help with estimating the hyperparameters.



Outline

Logistic Model

Crowder (1978) reported an experiment on germinating seeds. The data set is a 2×2 factorial layout with

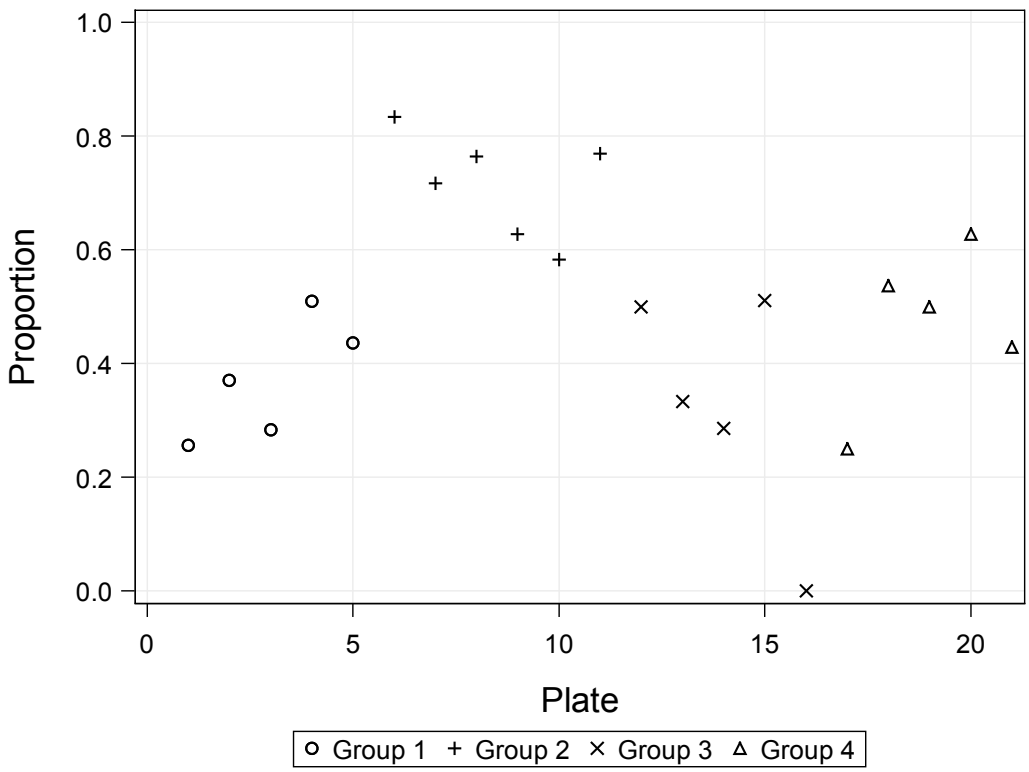
- Two types of seeds
- Two root extracts

The experiment included five or six replicates for each combination of seeds and root extracts.

A subset of the data:

r	n	seed	extract	ind
10	39	0	0	1
23	62	0	0	2
23	81	0	0	3
26	51	0	0	4
17	39	0	0	5
5	6	0	1	6
53	74	0	1	7
55	72	0	1	8
32	51	0	1	9
...				

Visualizing the Data Set



Logistic Regression

A natural way to model proportion data is to use the logistic regression with normal prior on the coefficients:

$$\begin{aligned} r_i &\sim \text{binomial}(n_i, p_i) \\ \mu_i &= \beta_0 + \beta_1 \cdot \text{seed}_i + \beta_2 \cdot \text{extract}_i + \beta_3 \cdot \text{seed}_i \cdot \text{extract}_i \\ \text{logit}(p_i) &= \log\left(\frac{p_i}{1-p_i}\right) = \mu_i \\ \pi(\beta_0, \beta_1, \beta_2, \beta_3) &\propto \text{normal}(0, \text{sd} = 1000) \end{aligned}$$

where $i = \{1, \cdots, 21\}$.

Fitting Logistic Regression in PROC MCMC

```
proc mcmc data=seeds outpost=postout1 seed=332786 nmc=20000
   stats=(summary intervals) diag=none;
   parms beta0-beta3;
   prior beta: ~ normal(0, sd=1000);
   mu = beta0 + beta1*seed + beta2*extract + beta3*seed*extract;
   pi = logistic(mu);
   model r ~ binomial(n = n, p = pi);
run;
```

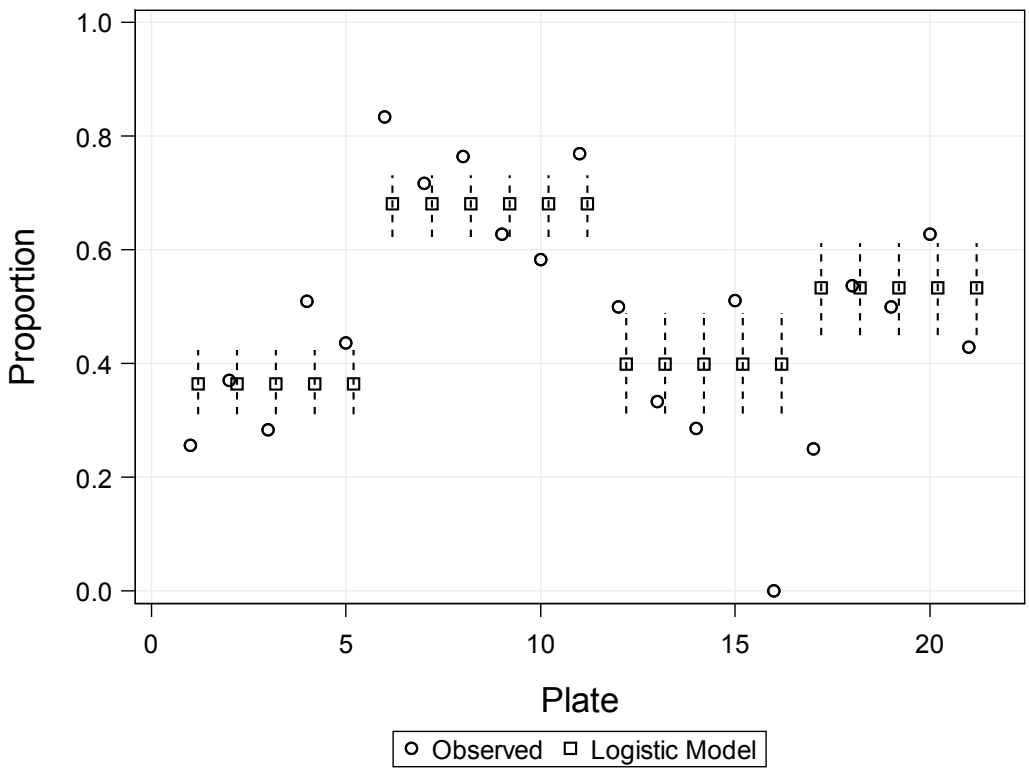
logistic : $pi = \frac{\exp(\mu)}{1+\exp(\mu)}$

Posterior Summary and Interval Statistics

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
beta0	20000	-0.5596	0.1256	-0.6457	-0.5583	-0.4704
beta1	20000	0.1444	0.2250	-0.00375	0.1458	0.2933
beta2	20000	1.3190	0.1792	1.1988	1.3189	1.4354
beta3	20000	-0.7723	0.3107	-0.9786	-0.7727	-0.5592

Posterior Intervals					
		Equal-Tail Interval		HPD Interval	
Parameter	Alpha				
beta0	0.050	-0.8175	-0.3210	-0.7989	-0.3059
beta1	0.050	-0.3015	0.5822	-0.3043	0.5763
beta2	0.050	0.9672	1.6852	0.9597	1.6752
beta3	0.050	-1.3891	-0.1716	-1.3474	-0.1414

Fit of the Logistic Model



Probit Model

You can change from a logistic to a probit regression:

$$\begin{aligned} y_i|p_i &\sim \text{binomial}(n_i, p_i) \\ \mu_i &= \beta_0 + \beta_1 \cdot \text{seed}_i + \beta_2 \cdot \text{extract}_i + \beta_3 \cdot \text{seed}_i \cdot \text{extract}_i \\ p_i &= \Phi(\mu_i) \\ \pi(\beta_0, \beta_1, \beta_2, \beta_3) &\propto \text{normal}(0, \text{sd} = 1000) \end{aligned}$$

```
proc mcmc data=seeds outpost=postout1 seed=332786 nmc=20000
   stats=(summary intervals) diag=none;
   parms beta0-beta3;
   prior beta: ~ normal(0, sd=1000);
   mu = beta0 + beta1*seed + beta2*extract + beta3*seed*extract;
   pi = cdf("normal", mu, 0, 1);
   model r ~ binomial(n = n, p = pi);
run;
```


Poisson Model

Or a Poisson model, if the response variable is count data:

$$\begin{aligned} y_i|\lambda_i &\sim \text{poisson}(\lambda_i) \\ \mu_i &= \beta_0 + \beta_1 \cdot \text{seed}_i + \beta_2 \cdot \text{extract}_i + \beta_3 \cdot \text{seed}_i \cdot \text{extract}_i \\ \lambda_i &= \exp(\mu_i) \\ \pi(\beta_0, \beta_1, \beta_2, \beta_3) &\propto \text{normal}(0, \text{sd} = 1000) \end{aligned}$$

```
proc mcmc data=seeds outpost=postout1 seed=332786 nmc=20000
   stats=(summary intervals) diag=none;
   parms beta0-beta3;
   prior beta: ~ normal(0, sd=1000);
   mu = beta0 + beta1*seed + beta2*extract + beta3*seed*extract;
   lambda = exp(mu);
   model y ~ poisson(lambda);
run;
```

Outline

Random-Effects Model

Recall that in the `trials` analysis, we considered a simple model:

$$\begin{aligned} \text{event}_i &\sim \text{binomial}(n_i, p) \\ p &\sim \text{beta}(a, b) \end{aligned}$$

which assumes that all groups share the same characteristic (success or failure probability).

Random-effects models enable you to model group-specific characteristics, such as different trials share similar but different failure probabilities:

$$\begin{aligned} \text{event}_i &\sim \text{binomial}(n_i, p_i) \\ p_i &\sim \text{prior} \end{aligned}$$

A Typical Random-Effects Model

A generic setup of a random-effects model:

$$Y_{ij} = \alpha \cdot X_{ij} + \beta_j + \epsilon_{ij}, \quad j = 1 \cdots J, \quad i = 1 \cdots n_j \tag{1}$$

where

- Y_{ij} is the response value of the ith subject in the jth cluster
- J is the total number of clusters, and n_j is the total number of subjects in the jth cluster.
- α is the fixed-effects parameter for X_{ij}
- ϵ_{ij} are the i.i.d. errors from a common distribution
- β_j is the varying intercepts

Often it is assumed that β_j arise from the same distribution,

$$\beta_j \sim \pi(\theta) \tag{2}$$

where θ are the hyperparameters.

Different Types of Random-Effects Models

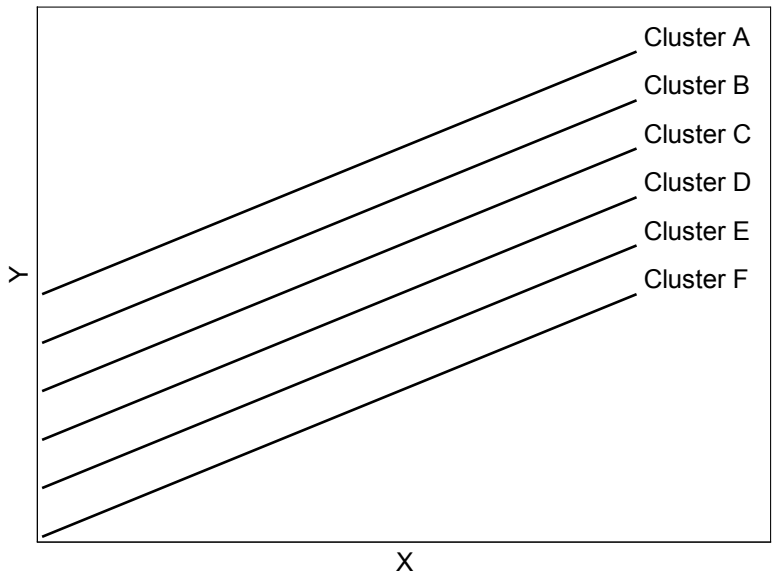
- If ϵ_{ij} in model (1) is assumed to have a normal distribution, then the model becomes a linear random-effects model.
- If you choose to model

$$E(Y_{ij}) = g(\alpha \cdot X_{ij} + \beta_j)$$

  where Y_{ij} is assumed to arise from the exponential family and $g(\cdot)$ is a one-to-one monotone transformation, then the model becomes a generalized linear random-effects model.
- If Y_{ij} relates to the regression via nonlinear transformation, the model becomes a more general nonlinear random-effects model.

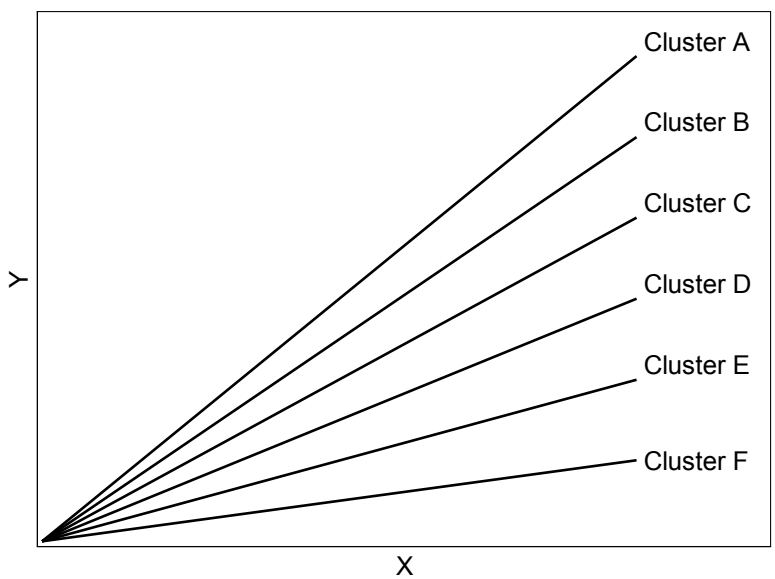
Varying-Intercept Models

$$Y_{ij} = \alpha \cdot X_{ij} + \beta_j$$



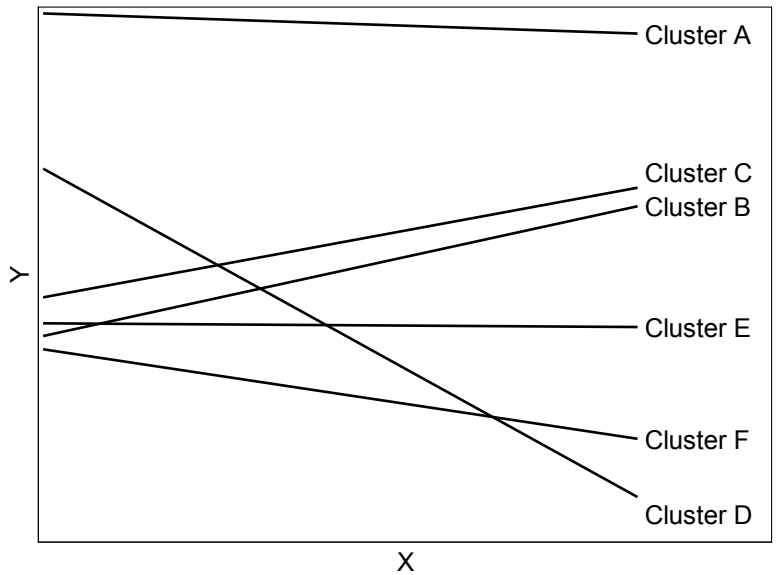
Varying-Slope Models

$$Y_{ij} = \gamma_j \cdot X_{ij} + C$$



Varying-Intercept and Varying-Slope Models

$$Y_{ij} = \gamma_j \cdot X_{ij} + \beta_j$$



The RANDOM Statement

The RANDOM statement is designed to construct random-effects models in PROC MCMC. The statement

- specifies the random-effects parameters (β_j, γ_j, and so on).
- makes the following conditional independence assumption:

$$\begin{aligned} \beta_j &\sim \pi(\theta) \\ \beta_i &\perp \beta_j \quad \textit{a priori} \end{aligned}$$

where θ are the hyperparameters.

Syntax

RANDOM *random-effect* ~ distribution SUBJECT= <options> ;

random-effect : defines the effect

distribution : specifies its prior distribution (beta/binary/gamma/igamma/laplace/normal/mvn/general)

SUBJECT= : identifies group membership

options : control initial values, monitoring list, and so on

Random Effects

- A program can have multiple RANDOM statements:
  - one for the classroom-level
  - one for the school-level
  - ...
- You can fit nested or nonnested models:
  - nested models: levels of one factor must cluster within the levels of another factor, such as students clustered within classrooms.
    - ★ the classroom effect can be the hyperparameters to the student effects
  - nonnested models: levels of factors can cross, such as a student effect and an age effect;
- The effects can enter the model in any linear or nonlinear form
  - $\beta_j + \gamma_k$
  - $\exp(\beta_j)$
  - ...

The SUBJECT= Variable

This is a data set variable that indicates clustering of the random effects.

- The number of random-effects parameters in a RANDOM statement is determined by the number of unique values in the SUBJECT= variable.
- The SUBJECT= variable be numeric or character, and doesn't have to be sorted

1	27513	07/13/1995	John
1	27513	01/31/2003	Mary
2	01440	10/12/1997	Ken
2	17923	08/03/2010	John
...	...	...	...

Understand the SUBJECT= Syntax

y	x	gender	group
0	13	female	1
1	10	male	2
1	11	female	3
0	7	male	4
0	10	female	5

We want to fit a model with two random effects, gender (α_j) and group (β_k), that has this general form:

$$\begin{aligned} \mu_i &= g(\alpha_j \cdot x_i + \beta_k) \\ y_i &\sim \text{dist}(\mu_i) \end{aligned}$$

where $i = \{1, \cdots, 5\}$, $j = \{1, 2\}$, and $k = \{1, \cdots, 5\}$.

Understand the SUBJECT= Syntax

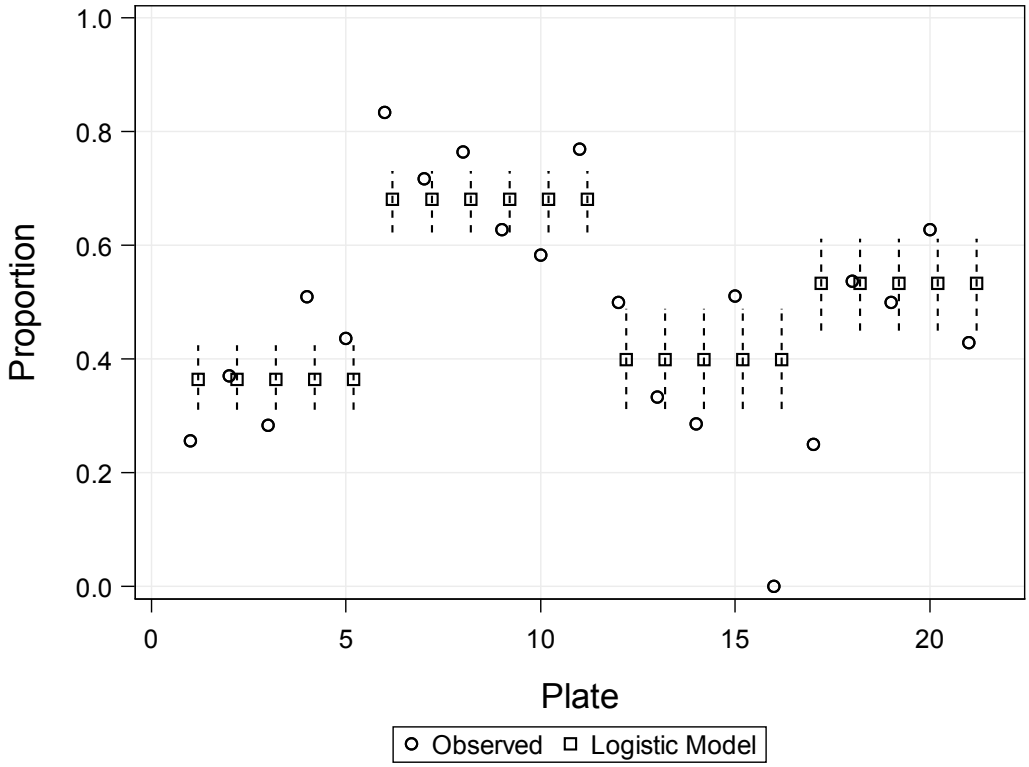
```
random alpha ~ dist() subject=gender;
random beta  ~ dist() subject=group;
mu = g(alpha * x + beta);
model y ~ dist(mu);
```

PROC MCMC internally creates two α parameters (α_{female}, α_{male}), five β parameters ($\beta_1, \cdots, \beta_5$), and interprets the input data set as:

y	x	gender	group	equation processed
0	13	α_{female}	β_1	$\mu = g(\alpha_{\text{female}} \cdot 13 + \beta_1)$
1	10	α_{male}	β_2	$\mu = g(\alpha_{\text{male}} \cdot 10 + \beta_2)$
1	11	α_{female}	β_3	$\mu = g(\alpha_{\text{female}} \cdot 11 + \beta_3)$
0	7	α_{male}	β_4	$\mu = g(\alpha_{\text{male}} \cdot 7 + \beta_4)$
0	10	α_{female}	β_5	$\mu = g(\alpha_{\text{female}} \cdot 10 + \beta_5)$

Outline

Recall the Logistic Example



Excessive Variation

Crowder (1978) noted heterogeneity of the proportions between replicates. To account for excessive variation, Brewslow and Clayton (1993) suggested a random-effects logistic regression:

$$
\begin{aligned}
r_i &\sim \text{binomial}(n_i, p_i) \\
\mu_i &= \beta_0 + \beta_1 \cdot \text{seed}_i + \beta_2 \cdot \text{extract}_i + \beta_3 \cdot \text{seed}_i \cdot \text{extract}_i \\
p_i &= \text{logistic}(\mu_i + \delta_i) \\
\pi(\beta_0, \beta_1, \beta_2, \beta_3) &\propto \text{normal}(0, \text{sd} = 1000) \\
\delta_i &\sim \text{normal}(0, \text{var} = \sigma^2) \\
\sigma^2 &\sim \text{igamma}(\text{shape} = 0.01, \text{scale} = 0.01)
\end{aligned}
$$

where δ_i is the random-effects parameter, and σ^2 is the hyperparameter variance.

Random-Effects Logistic Regression

The following program fits a logistic random-effects model:

```
proc mcmc data=seeds outpost=postout seed=332786 nmc=20000
   stats=(summary intervals) diag=none;
   parms beta0-beta3 s2 1;
   prior beta: ~ normal(0, sd=1000);
   prior s2 ~ igamma(0.01, s=0.01);
   mu = beta0 + beta1*seed + beta2*extract + beta3*seed*extract;
   random delta ~ normal(0, var=s2) subject=_obs_;
   pi = logistic(mu + delta);
   model r ~ binomial(n = n, p = pi);
run;
```

subject=_obs_ : fits observational level random effects

Model Parameters Information

Parameters				
Block	Parameter	Sampling Method	Initial Value	Prior Distribution
1	s2	Conjugate	0.00990	igamma(0.01, s=0.01)
2	beta0	N-Metropolis	0	normal(0, sd=1000)
	beta1		0	normal(0, sd=1000)
	beta2		0	normal(0, sd=1000)
	beta3		0	normal(0, sd=1000)

Random Effect Parameters					
Parameter	Sampling Method	Subject	Number of Subjects	Subject Values	Prior Distribution
delta	N-Metropolis	_OBS_	21	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 ...	normal(0, var=s2)

Posterior Inference

By default, PROC MCMC does not display posterior estimates of the random-effects parameters (there could be too many):

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles		
				25%	50%	75%
beta0	20000	-0.5519	0.1943	-0.6799	-0.5523	-0.4259
beta1	20000	0.0986	0.3167	-0.1026	0.1119	0.3075
beta2	20000	1.3606	0.2815	1.1844	1.3580	1.5385
beta3	20000	-0.8785	0.4589	-1.1652	-0.8824	-0.5905
s2	20000	0.1239	0.1099	0.0528	0.0968	0.1619

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
beta0	0.050	-0.9347	-0.1547	-0.9525	-0.1865
beta1	0.050	-0.5636	0.7045	-0.4972	0.7486
beta2	0.050	0.8072	1.9280	0.8089	1.9284
beta3	0.050	-1.7696	0.0271	-1.7329	0.0615
s2	0.050	0.00895	0.3964	0.00221	0.3180

Monitoring Random-Effects Parameters

The MONITOR= option enables you to display

- all of these random-effects estimates:

```
random delta ~ n(0, var=s2) subject=_obs_ monitor=(delta);
```

- a subset of these estimates:

```
random delta ~ n(0, var=s2) subject=_obs_ monitor=(1 2 6);
```

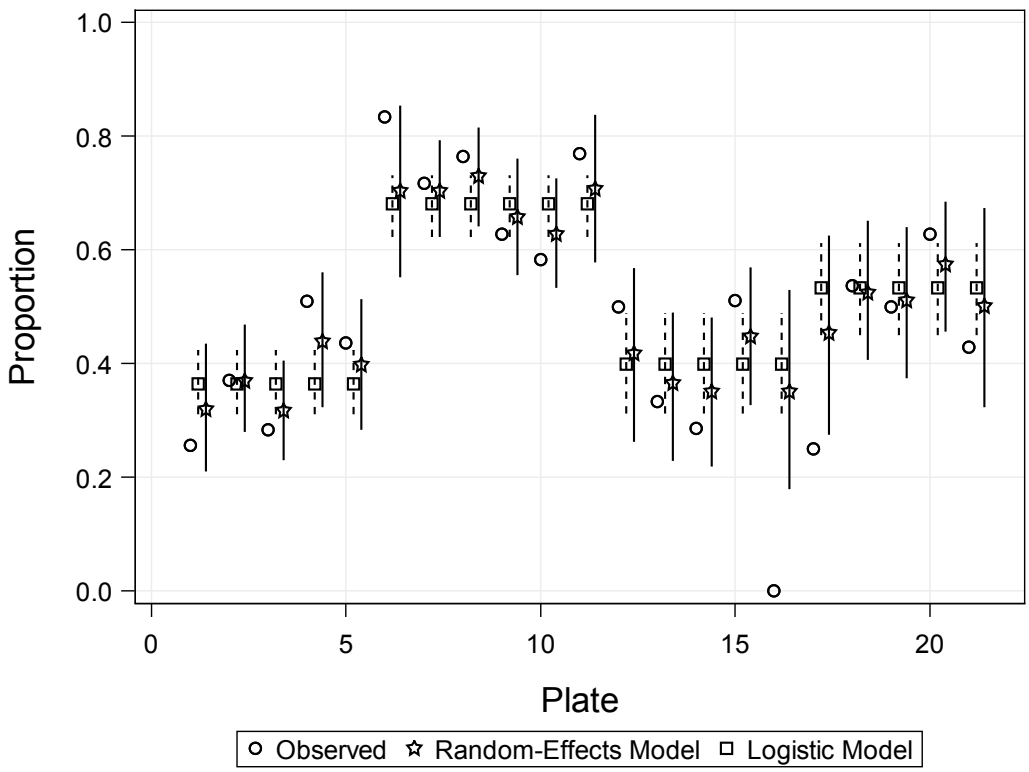
- have the procedure choose a subset:

```
random delta ~ n(0, var=s2) subject=_obs_ monitor=(random(3));
```

Monitored Random-Effects Parameters

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
beta0	20000	-0.5519	0.1943	-0.6799	-0.5523	-0.4259
beta1	20000	0.0986	0.3167	-0.1026	0.1119	0.3075
beta2	20000	1.3606	0.2815	1.1844	1.3580	1.5385
beta3	20000	-0.8785	0.4589	-1.1652	-0.8824	-0.5905
s2	20000	0.1239	0.1099	0.0528	0.0968	0.1619
delta_7	20000	0.0644	0.2403	-0.0809	0.0595	0.2175
delta_8	20000	0.1972	0.2508	0.0301	0.1821	0.3555
delta_19	20000	-0.00081	0.2869	-0.1650	-0.00755	0.1563

Fit of the Random-Effects Model



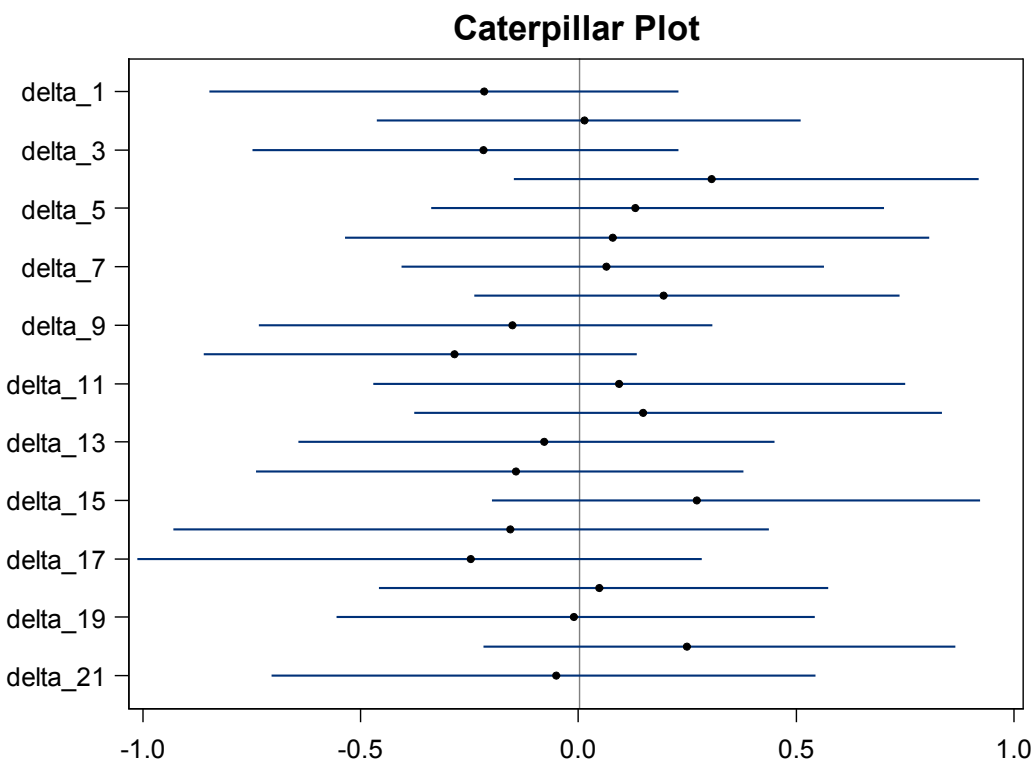
Caterpillar Plot

- A side-by-side bar plot of the 95% equal-tail posterior intervals for multiple parameters
- Useful in visualizing and comparing parameters.
- Better than overlay kernel density plots.

The %CATER autocall macro creates a caterpillar plot:

```
%cater(data=postout, var=delta_:);
```


Caterpillar Plot



Outline

Hyperprior in Random-Effects Models

Back to the `trials` analysis model:

$$\text{event}_i \sim \text{binomial}(n_i, p_i)$$

where i indexes the group.

The group-specific p_i is a weighted average of the pooled estimate (shared p) and independent estimates (seperate analysis). The amount of shrinkage is determined by the hyperprior distribution.

Here we consider two common choices:

$$\begin{aligned} p_i &\sim \text{beta}(a, b) \\ \text{logit}(p_i) &\sim \text{normal}(\mu, \sigma^2) \end{aligned}$$

Hyperparameters

If you choose constant values for a, b, or σ^2, you decide *a priori* the amount of shrinkage you want on the p_i. For example:

- Choosing $a = 1$ and $b = 1$, or $\sigma^2 = \infty$, implies no shrinkage on the p_i. The random-effects model becomes an independent model.
- Choosing $\sigma^2 = 0$ imposes no variation amongst p_i. This reduces the random-effects model to the pooled model.

Hyperparameters

Empirical Bayes offers one way of choosing these hyperparameters.

- find estimates a, b, μ, or σ^2 by maximizing the posterior marginal distributions of $\pi(a, b|\mathbf{x}, \mathbf{y})$ or $\pi(\mu, \sigma^2|\mathbf{x}, \mathbf{y})$
- plug in these estimates as the hyperparameters

This provides reasonable inferences if there are enough units or groups in the data to estimate the variance.

But the plug-in approach ignores uncertainty that your data indicates about the amount of shrinkage that should be used.

Hyperprior Distributions

Ideally, the data should decide the right amount of strength you want to borrow from different groups in the analysis. This amounts to placing hyperprior distributions on the hyperparameters.

For example, see Spiegelhalter, Abrams, and Myles (2004), and Gelman et al. (2003) for discussions on how to select such prior distributions. Strategies include:

- noninformative
- elicitation
- summary of evidence

Hyperprior Distributions

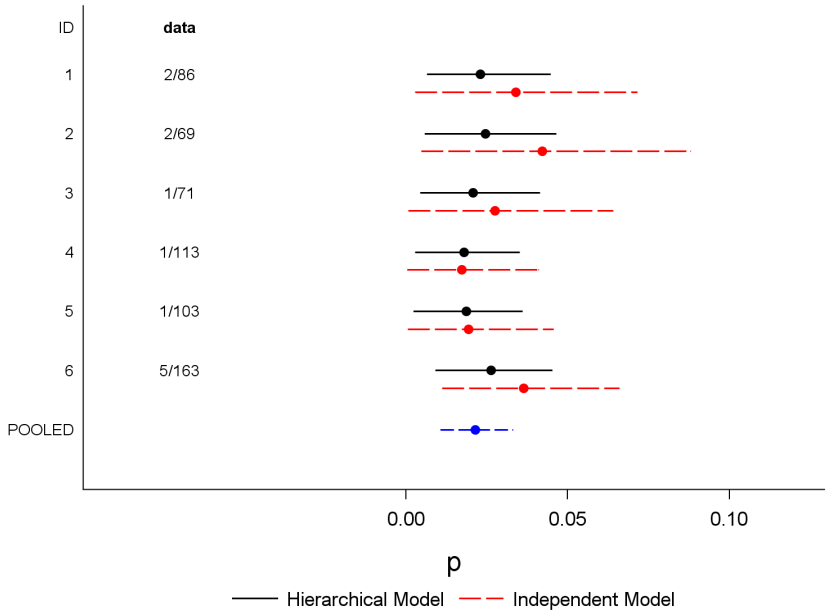
First, let's consider the beta hyperprior model and use proper but diffuse prior distributions on a and b:

$$\begin{aligned} \text{event}_i &\sim \text{binomial}(n_i, p_i) \\ p_i &\sim \text{beta}(a, b) \\ a,\ b &\sim \text{exponential}(\text{scale} = 100) \end{aligned}$$

```
proc mcmc data=trials nmc=50000 outpost=outm seed=17;
   parm a b;
   prior a b ~ expon(scale=100);
   random p ~ beta(a, b) subject=center;
   model event ~ binomial(n, p);
   run;
```

Posterior Estimates of Probabilities

Compare to Page 217



95% HPD intervals and estimates for p_i. The solid line is the random-effects model; the dashed line is the independence model (individual analysis); the bottom line is the overall (pooled) model.

Modeling σ^2 in Hierarchical Models

Secondly, we consider the following model:

$$\begin{aligned} \text{event}_i &\sim \text{binomial}(n_i, p_i) \\ \gamma_i = \text{logit}(p_i) &\sim \text{normal}(\mu, \sigma^2) \\ \mu &\sim \text{normal}(0, \text{precision} = 10^{-6}) \end{aligned}$$

What type of (noninformative) prior distribution should be used on σ^2?

Some Frequently used Prior Distributions for σ^2

- The Jeffreys' prior is a popular choice for the variance parameter in a normal model:

$$\pi(\sigma^2) \propto \frac{1}{\sigma^2}$$

which is equivalent to a uniform prior on $\log(\sigma^2)$

```
prior s2 ~ general(-log(s2), lower=0);
```

BUT, this leads to an improper posterior distribution in the random-effects model and SHOULD NOT BE USED.

- Uniform on variance:

$$\pi(\sigma^2) \propto 1$$

```
prior s2 ~ general(0, lower=0);
```

When there are few groups, $\pi(\sigma) \propto 1$ if often recommended.

Some Frequently used Prior Distributions for σ^2

- Conjugate prior:

$$\pi(\sigma^2) \propto i\Gamma(\text{shape}{=}\alpha, \text{scale}{=}\beta)$$

```
prior s2 ~ igamma(shape=, scale=);
```

- $$\pi(\sigma^2) \propto i\Gamma(\epsilon, \epsilon) \Leftrightarrow \pi(\tau = \frac{1}{\sigma^2}) \propto \Gamma(\epsilon, \text{iscale} = \epsilon)$$

with $\epsilon = 0.001$ being used frequently. This is a prior that "mimics" the Jeffreys' prior (However, it can be highly influential) .

For more detailed discussion on noninformative prior selections on σ^2, see Gelman (2006, *Bayesian Analysis* 1:515).

Fitting the Model in PROC MCMC

For illustrative purposes, consider

$$\sigma^2 \sim \text{igamma}(0.001, \text{scale} = 0.001)$$

```
proc mcmc data=trials nmc=50000 outpost=outmc seed=17;
   parms mu s2;
   prior mu ~ n(0, prec=1e-6);
   prior s2 ~ igamma(0.001, s=0.001);
   random gamma ~ n(mu, var=s2) subject=center;
   logitP = logistic(gamma);
   model event ~ binomial(n,logitP);
   run;
```

logistic : $logitP = \frac{\exp(\gamma)}{1+\exp(\gamma)}$

Posterior Estimates

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
mu	50000	-3.9977	0.3544	-4.2103	-3.9641	-3.7630
s2	50000	0.1460	0.5927	0.00578	0.0239	0.1060
gamma_1	50000	-3.9709	0.3923	-4.2067	-3.9450	-3.7151
gamma_2	50000	-3.9452	0.4051	-4.1859	-3.9312	-3.6923
gamma_3	50000	-4.0280	0.4235	-4.2615	-3.9840	-3.7524
gamma_4	50000	-4.0751	0.4441	-4.2995	-4.0066	-3.7916
gamma_5	50000	-4.0667	0.4266	-4.2979	-4.0098	-3.7870
gamma_6	50000	-3.8935	0.3645	-4.1182	-3.8869	-3.6530

Transform γ_i to p_i Parameters

```
proc mcmc data=trials nmc=50000 outpost=outmc seed=17
   monitor=(mu s2 p);
   array p[6];
   parms mu s2;
   prior mu ~ n(0, prec=1e-6);
   prior s2 ~ igamma(0.001, s=0.001);
   random gamma ~ n(mu, var=s2) subject=center;
   logitP = logistic(gamma);
   model event ~ binomial(n, logitP);
   p[center] = logitP;
   run;
```

ARRAY : allocates an array to store p_i, which are functions of γ_i

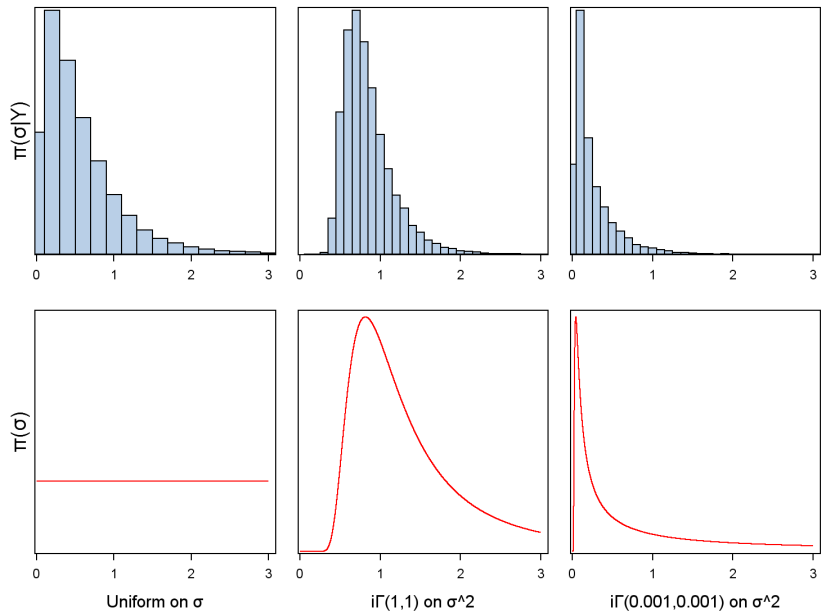
MONITOR : outputs model parameters and all elements of the array p

p : p[center] saves the correct transformation for each γ_i according to its cluster/center membership.

Posterior Estimates of the p_i Parameters

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
mu	50000	-3.9977	0.3544	-4.2103	-3.9641	-3.7630
s2	50000	0.1460	0.5927	0.00578	0.0239	0.1060
p1	50000	0.0198	0.00757	0.0147	0.0190	0.0238
p2	50000	0.0205	0.00842	0.0150	0.0192	0.0243
p3	50000	0.0189	0.00727	0.0139	0.0183	0.0229
p4	50000	0.0181	0.00677	0.0134	0.0179	0.0221
p5	50000	0.0182	0.00683	0.0134	0.0178	0.0222
p6	50000	0.0212	0.00761	0.0160	0.0201	0.0253

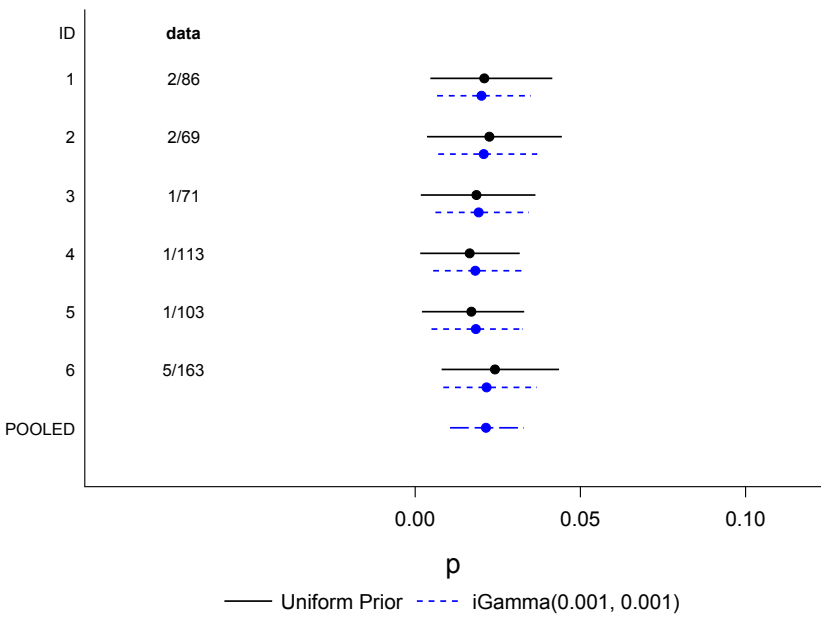
Prior Matters!



Top panels are $\pi(\sigma|\mathbf{y})$ based on three different prior distributions: uniform on σ, $i\Gamma(1,1)$ and $i\Gamma(0.001, 0.001)$ on σ^2. The popular choice of inverse-Gamma distribution is hardly noninformative.

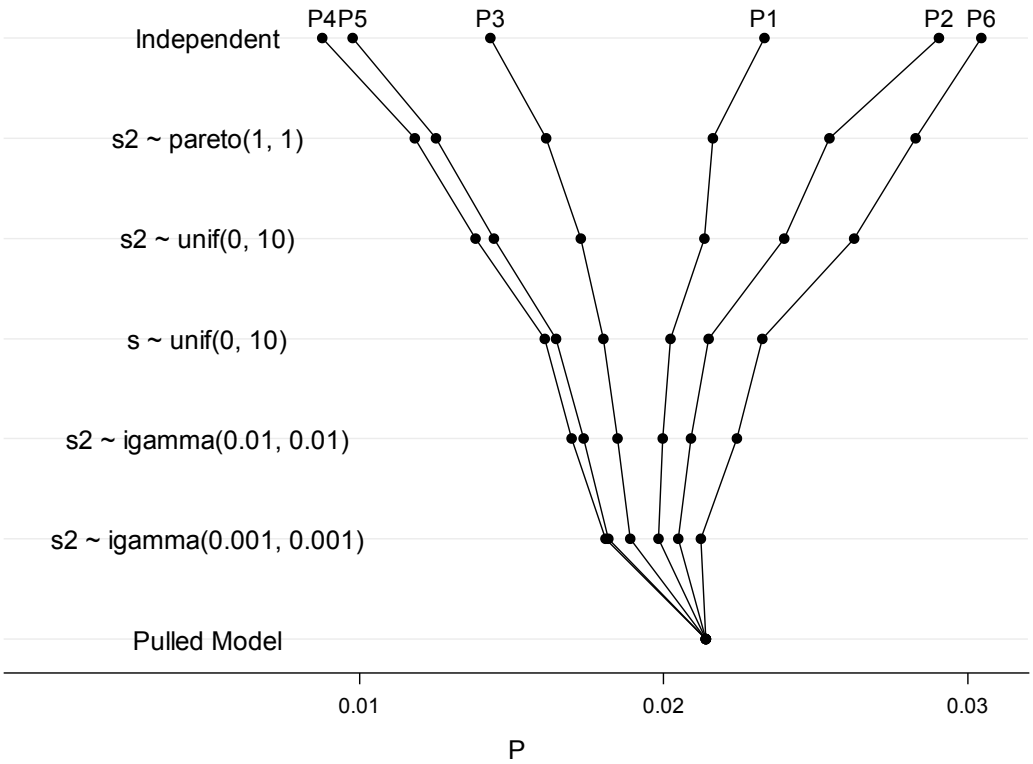
Posterior Estimates of Probabilities

◂ Compare to Page 208



95% HPD credible intervals and posterior point estimates for each p_i. There is an excessive amount of shrinkage using the $i\Gamma(0.001, 0.001)$ prior.

Shrinkage Effects



Outline

Repeated Measurements

- Individual subjects have repeated observations, for example, over time (longitudinal), or within subjects (test scores).
- Different from time series data in the sense that the number of measurements per subject is generally not very large
- Covariates information is avaliable either at subject or measurement level.
- Subjects can have same number of repeated measures (balanced) or uneven number of measures (unbalanced)

Two-Arm Study

- The data comes from a two arms (control vs treatment) experiment that is carried out at eight sites.
- The response variables are success counts, y_c and y_c, out of the same number of trials, $n_c = 132$ and $n_t = 148$, respectively.
- Each site sees different numbers of repeats (from 1 up to 13).
- Additional covariates information is withheld (but you can easily add them).

Control Data ($n_c = 132$)

id	yc1	yc2	yc3	yc4	yc5	yc6	yc7	yc8	yc9	yc10	yc11
1	40	26	0	1	20	.	.	.	.	.	.
2	2	0	0	10	1	7	19	.	.	.	.
3	2	.	.	.	.	.	.	.	.	.	.
4	2	.	.	.	.	.	.	.	.	.	.
5	0	2	43	.	.	.	.	.	.	.	.
6	1	2	2	8	20	1	8	14	1	1	1
7	0	0	1	3	2	1	2	2	2	.	.
8	2	14	.	.	.	.	.	.	.	.	.

Treatment Data ($n_t = 148$)

id	yt1	yt2	yt3	yt4	yt5	yt6	yt7	yt8	yt9	yt10	yt11
1	57	34	2	3	27	.	.	.	.	.	.
2	7	2	2	24	3	2	19	.	.	.	.
3	3	.	.	.	.	.	.	.	.	.	.
4	0	.	.	.	.	.	.	.	.	.	.
5	2	2	75	.	.	.	.	.	.	.	.
6	4	4	1	13	28	2	13	15	3	2	3
7	4	2	2	13	6	8	4	0	1	.	.
8	0	18	.	.	.	.	.	.	.	.	.

Input data is in a vector format.

id	rep	yc	yt
1	1	40	57
1	2	26	34
1	3	0	2
1	4	1	3
1	5	20	27
2	1	2	7
2	2	0	2
2	3	0	2
2	4	10	24
2	5	1	3
2	6	7	2
2	7	19	19
3	1	2	3
4	1	2	0
5	1	0	2
5	2	2	2
5	3	43	75
6	1	1	4
...			

Simple Model

For every $i = \{1, \cdots, 39\}$

$$\begin{aligned} y_{ci} &\sim \text{binomial}(n_c, p_c) \\ \text{logit}(p_c) &= \gamma \\ y_{ti} &\sim \text{binomial}(n_t, p_t) \\ \text{logit}(p_t) &= \gamma + \theta \\ \gamma, \theta &\sim \text{normal}(0, sd = 10) \end{aligned}$$

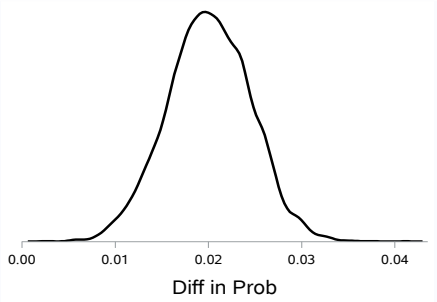
where γ is the baseline (for control group) and θ is the treatment effect (log of odds ratio).

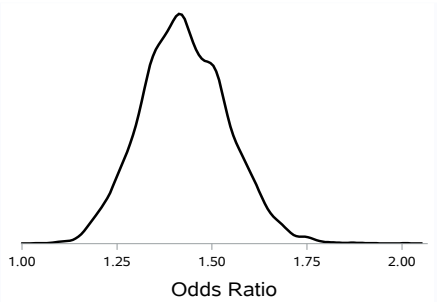
Some quantities of interest could be:

- Difference in probabilities: p_t - p_c
- Odds ratio: $\exp(\theta)$

Fitting the Population Model

```
proc mcmc data=TwoArms nmc=20000 seed=1
   monitor=(p_diff or);
   parm gamma theta;
   prior gamma theta ~ n(0, sd=10);

   pc = logistic(gamma);
   model yc ~ binomial(132, pc);
   pt = logistic(gamma + theta);
   model yt ~ binomial(148, pt);

   or = exp(theta);
   p_diff = pt - pc;
run;
```





Observational-level Model

For every $i = \{1, \cdots, 39\}$

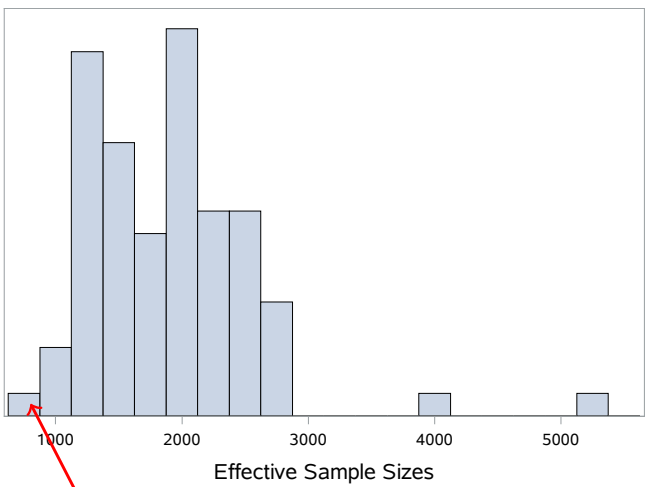
$$
\begin{aligned}
y_{ci} &\sim \text{binomial}\left(n_c, p_{ci}\right) \\
\text{logit}(p_{ci}) &= \gamma_i \\
\gamma_i &\sim \text{normal}(\mu_\gamma, \tau_\gamma) \\
y_{ti} &\sim \text{binomial}\left(n_t, p_{ti}\right) \\
\text{logit}(p_{ti}) &= \gamma_i + \theta_i \\
\theta_i &\sim \text{normal}(\mu_\theta, \tau_\theta) \\
\mu_\gamma,\ \mu_\theta &\sim \text{normal}(0, sd = 10) \\
\tau_\gamma,\ \tau_\theta &\sim \Gamma(3, iscale = 1)
\end{aligned}
$$

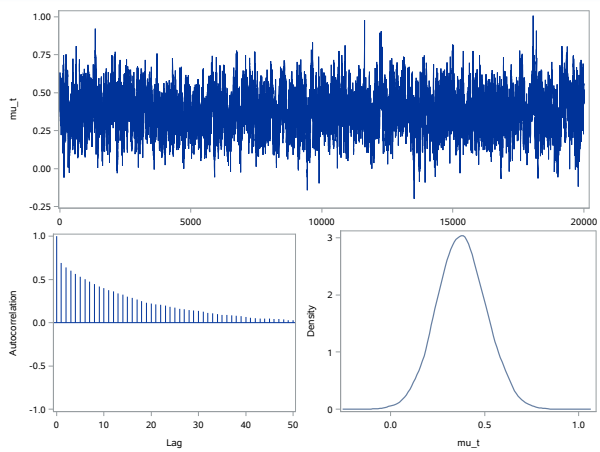
The number of parameters jumps from 2 to 82.

Fitting the Observational-Level Model

```
proc mcmc data=TwoArms nmc=20000 seed=1 outpost=ObsOut;
   parms mu_g mu_t tau_g tau_t;

   prior mu_g mu_t ~ n(0, prec=0.1);
   prior tau_g tau_t ~ gamma(3, iscale=1);

   random gamma ~ n(mu_g, prec=tau_g) subject=_obs_;
   random theta ~ n(mu_t, prec=tau_t) subject=_obs_;

   pc = logistic(gamma);
   model yc ~ binomial(132, pc);
   pt = logistic(gamma + theta);
   model yt ~ binomial(148, pt);
run;

%ess(data=ObsOut, var=mu: tau: gamma: theta:, out=ess);
```


ESSs of All Parameters

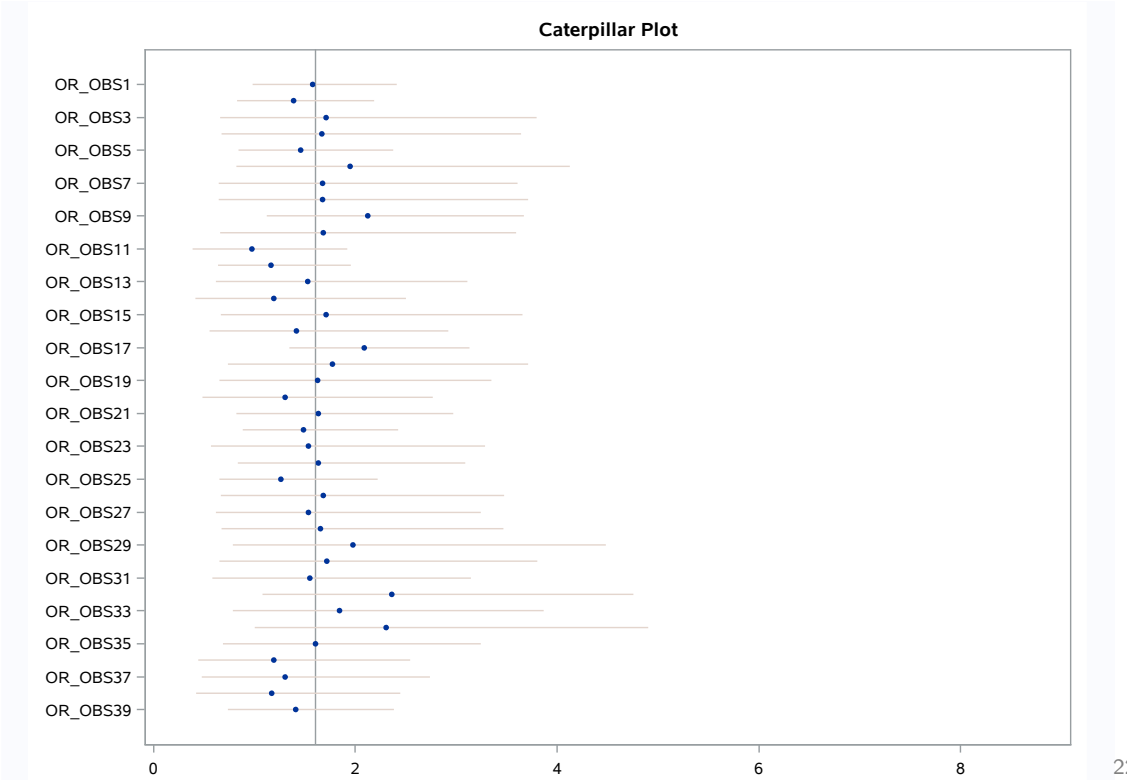




μ_t, smallest ESS of 823.

Observational-level odds ratios are rather similar to each other, indicating maybe an over-simplifying assumption on the random effects.

compare to page 231



Multi-level Model

It is more realistic to consider the site-effect in the model, where the repeated measures are modelled as similar to each other within each site:

$$
\begin{aligned}
y_{ci} &\sim \text{binomial}\left(n_c, p_{ci}\right) \\
\text{logit}(p_{ci}) &= \gamma_i \\
\gamma_{\{i,(j)\}} &\sim \text{normal}(\mu_{\gamma_j}, \tau_{\gamma_j}) \\
\mu_{\gamma_j} &\sim \text{normal}(\mu_\gamma, \tau_\gamma) \\
\mu_\gamma &\sim \text{normal}(0, sd = 10) \\
\tau_\gamma,\ \tau_{\gamma_j} &\sim \Gamma(3, iscale = 1)
\end{aligned}
\qquad
\begin{aligned}
y_{ti} &\sim \text{binomial}\left(n_t, p_{ti}\right) \\
\text{logit}(p_{ti}) &= \gamma_i + \theta_i \\
\theta_{\{i,(j)\}} &\sim \text{normal}(\mu_{\theta_j}, \tau_{\theta_j}) \\
\mu_{\theta_j} &\sim \text{normal}(\mu_\theta, \tau_\theta) \\
\mu_\theta &\sim \text{normal}(0, sd = 10) \\
\tau_\theta,\ \tau_{\theta_j} &\sim \Gamma(3, iscale = 1)
\end{aligned}
$$

where $i = \{1, \cdots, 39\}$ indexes observations, $j = \{1, \cdots, 8\}$ indexes sites, and $\{i, (j)\}$ indexes repeated measures in the jth site.

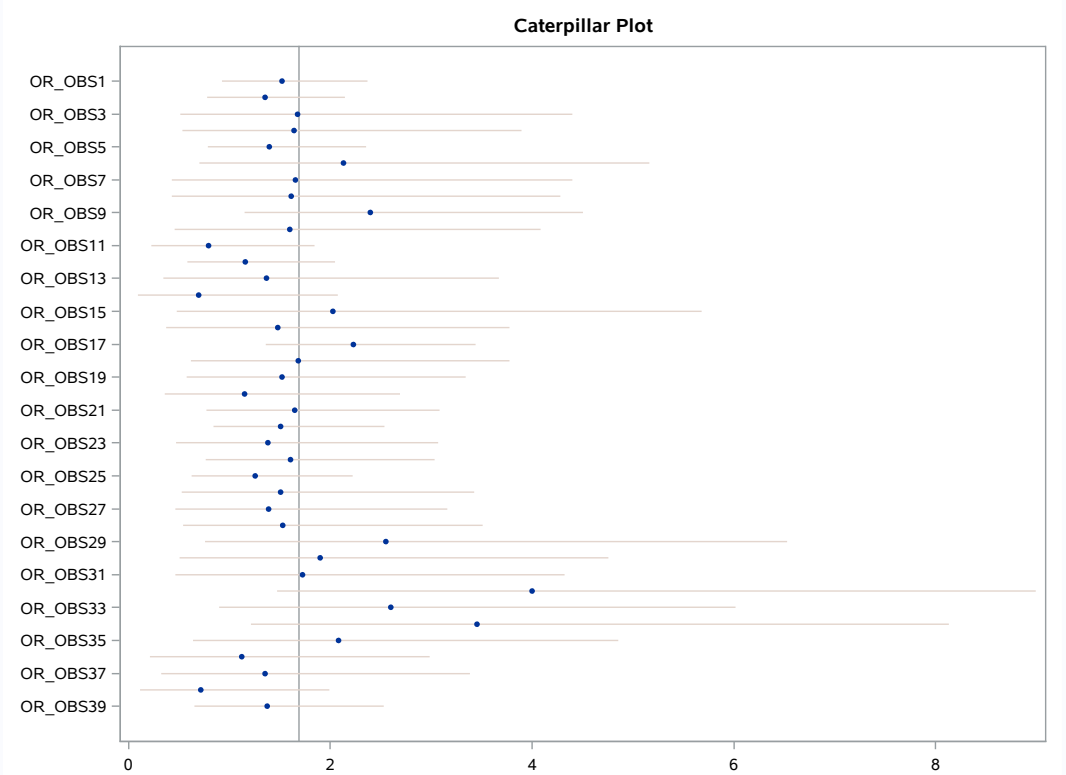
Fitting the Observational-Level Model

```
proc mcmc data=TwoArms nmc=20000 seed=1 outpost=MultOut;
   parms mu_g0 mu_t0 tau_g0 tau_t0;

   prior mu_g0 mu_t0 ~ n(0, prec=0.1);
   prior tau_g0 tau_t0 ~ gamma(3, iscale=1);

   random mu_g ~ n(mu_g0, prec=tau_g0) subject=id;
   random tau_g ~ gamma(shape=3, iscale=1) subject=id;
   random gamma ~ n(mu_g, prec=tau_g) subject=_obs_;

   random mu_t ~ n(mu_t0, prec=tau_t0) subject=id;
   random tau_t ~ gamma(shape=3, iscale=1) subject=id;
   random theta ~ n(mu_t, prec=tau_t) subject=_obs_;

   pc = logistic(gamma);
   model yc ~ binomial(132, pc);
   pt = logistic(gamma + theta);
   model yt ~ binomial(148, pt);
run;
```

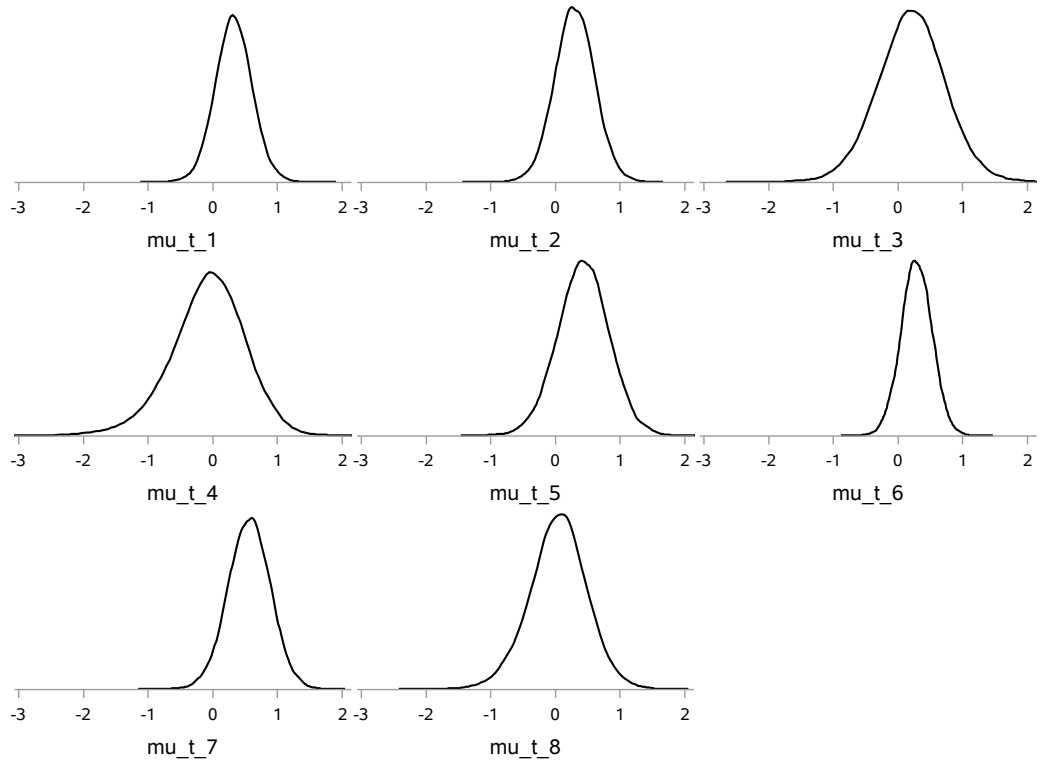
The SUBJECT= variables must be nested.

Odds Ratios Estiamtes by the Multi-level Model

compare to page 228



Posterior Densities of μ_{θ_j}



Outline

Introduction

- Missing data problems arise frequently in practice and are caused by many circumstances.
  - study subjects might fail to answer questions on a questionnaire,
  - data can be lost
  - covariate measurements might be unavailable
  - and so on...
- The impact of missing data on inference is potentially important, especially if subjects that have missing data differ systematically from those that have complete data.
- Coherent estimation and valid inference require adequate modeling of the missing values.

Bayesian Approach

Bayesian methods for missing data problems are straightforward.

- Bayesian paradigm treats any unknown quantities as random variables.
- Missing values are treated as additional variables that need to be estimated.
- The approach is very general and capable of handling complex missing data scenarios.

Notations

- Let $Y = \{Y_{\text{obs}}, Y_{\text{mis}}\}$ be the response variable of length n ($\{y_i\}$), where Y_{obs} and Y_{mis} denote the observed and missing values, respectively.
- The sampling distribution is assumed to have the generic form:

$$y_i \sim f(y_i|x_i, \theta)$$

  where $f(\cdot)$ is a known distribution (e.g. the likelihood), x_i are the covariates, and θ is the parameter of interest.
- Let $R_Y = (r_1, \cdots, r_n)$ be the missing value indicator, also called the missingness random variable, where $r_{y\,i} = 1$ if y_i is missing and $r_{y\,i} = 0$ otherwise. R is known when the Y are known.

Covariates Missing

Similar to missing in response variables, you can have missing covariates.

- Let $X = \{X_{\text{obs}}, X_{\text{mis}}\}$, and X can be multidimensional.
- Typically, covariates are considered to be fixed constants. But here, X is treated as a random variable:

$$x_i \sim \pi(x_i|u_i, \eta)$$

  where $\pi(\cdot)$ is a "prior" distribution. The model can have u_i as additional covariates and η the parameter of interest.
- Similarly, you have R_X the missing value indicator for each covariate X.

Objective

In the Bayesian approach, you can estimate the joint posterior distribution:

$$\pi(\theta, \eta, \mathbf{y}_{\text{mis}}, \mathbf{x}_{\text{mis}} | \mathbf{y}_{\text{obs}}, \mathbf{x}_{\text{obs}}, \mathbf{u})$$

The Monte Carlo in MCMC enables you to obtain the posterior marginal distributions for the parameters of interest:

$$\pi(\theta | \mathbf{y}_{\text{obs}}, \mathbf{x}_{\text{obs}}, \mathbf{u})$$
$$\pi(\eta | \mathbf{y}_{\text{obs}}, \mathbf{x}_{\text{obs}}, \mathbf{u})$$

Uncertainty about the missing values is fully propagated and incorporated in your inferences.

Classifications of Missing Data

Generally speaking, there are three types of missing data models (Rubin 1976):

- Missing Completely at Random
- Missing at Random
- Missing not at Random
  - selection model
  - pattern-mixture model

Missing Complete at Random

- **Missing Complete at Random (MCAR)** – if the failure to observe a value does not depend on any data, observed or missing.
  - The probability of observing a missing y_i is independent of other y_j, for $j \neq i$, and is independent of other covariates x_i.
- Under the MCAR assumption, you can use only the observed data in the analysis. This is called a **complete-case (CC)** analysis.
  - If the MCAR assumption fails to hold, a CC analysis is biased.
  - If the MCAR assumption holds, a CC analysis is unbiased but less efficient than an analysis that uses the full data.

Missing at Random

- **Missing at Random (MAR)** – if the failure to observe a value is independent of missing values but may depend on observed value.
- MCAR assumes that the observed quantities are no longer random samples and adjustments should be made accordingly (a more realistic assumption than MCAR).
- In MAR, the missing mechanism, (R_Y), does not need to be modeled and can be ignored.
- MAR is sometimes referred to as *ignorable missing*; it is not the missing values but the missing mechanism that can be ignored.

Missing not at Random

- **Missing not at Random (MNAR)** – if the failure to observe a value depends on unobserved observations (the would-have-been values).
- MNAR is the most general and most complex missing data scenario, and is frequently encountered in longitudinal studies with repeated measures.
  - In a Quality of Life (QOL) study, a patient can drop out depends on how sick they are, which is unobserved.
- The missing mechanism is no longer ignored and a model for R_Y is required. MNAR is sometimes referred to as *nonignorable missing*.

Modeling MNAR

In MNAR, you have a joint likelihood function over (R, Y):

$$f_{R,Y}(r, y|x, \theta)$$

- The **selection model** factors joint distribution into:

$$f(r, y|x, \theta) \propto f(y|x, \alpha) \cdot f(r|y, x, \beta)$$

  where $\theta = (\alpha, \beta)$.
  - $f(y|x, \alpha)$, known as the *outcome model*, is the typical likelihood.
  - $f(r|y, x, \beta)$: typically a binary model
- The **pattern-mixture model** factors the opposite way:

$$f(r, y|x, \theta) \propto f(y|r, x, \delta) \cdot f(r|x, \gamma)$$

  where $\theta = (\gamma, \delta)$.

Selection vs Pattern-Mixture

Some prefer the selection approach:

- It is a natural way of decomposing the joint distribution.
- In MAR analysis, you don't need to include R in the analysis.
- When MNAR analysis is required, adding the conditional model is easy.

Others prefer the pattern-mixture approach:

- The marginal model can model different patterns in R.
- You can build meaningful models for subsets of the response variable conditional on different missing patterns.
- On the other hand, you must always model R.

Handling of Missing Values in PROC MCMC

The MODEL statement handles the estimation of all missing values:

MODEL *variable-list* $\sim$ distribution / <options> ;

- The distribution is the usual likelihood function when the MODEL statement is applied to a response variable;
- It becomes a prior distribution for a covariate;

The procedure steps through the input data set, identifies all missing values that are in *variable-list*, creates a separate parameter for each missing value, and draw samples from their posterior distributions.

Handling of Missing Values in PROC MCMC

- PROC MCMC models missing values only for variables that are specified in the MODEL statement. For example, suppose that there are missing values in y:

```
MODEL y ~ normal(mu, var=1);
```

  Each missing value in y becomes a parameter and is sampled in the Markov chain.

- Records that contain missing values in other data set variables (that are not in the MODEL statement) are discarded. Suppose that there are missing values in x:

```
mu = beta0 + beta1 * x;
MODEL y ~ normal(mu, var=1);
```

  PROC MCMC does not model any missing values in x.

Options in the MODEL Statement

The options in the MODEL statement are avaliable only when there are missing values in the variables:

MODEL *variable-list* ~ distribution / <options> ;

Option	Description
INITIAL=	specifies the initial values of the missing data, which are used to start the Markov chain.
MONITOR=	outputs analysis for selected missing data variables.
NAMESUFFIX=	specifies how to create the names of the missing data variables.
NOOUTPOST	suppresses the output of the posterior samples of the missing data variables.

Bivariate Normal with Partial Missing

x1	1	1	−1	−1	2	2	−2	−2	.	.	.	.
x2	1	−1	1	−1	.	.	.	.	2	2	−2	−2

Table : Bivariate Normal Data with Partial Missing (Murray 1977)

The data are assumed to have zero means, and the parameter of interest is the covariance matrix $\mathbf{\Sigma}$ and correlation ρ. The likelihood is

$$\pi(\mathbf{x}|\boldsymbol{\mu}, \mathbf{\Sigma}) \propto \text{MVN}(\boldsymbol{\mu} = \mathbf{0}, \mathbf{\Sigma})$$

The prior on $\mathbf{\Sigma}$ is

$$\pi(\mathbf{\Sigma}) \propto \text{iWishart}(\nu = 3, S = \mathbf{I})$$

A CC analysis uses first four observations, which produces a rather simple estimate of ρ of 0. But this throws away all of the information that is in the other partially observed data.

Fitting MAR of the bivariate normal data with partial missing using PROC MCMC:

```
proc mcmc data=binorm nmc=20000 seed=17 outpost=pout
     monitor=(rho);
   array x[2] x1 x2;                          ! response variable
   array mu[2] (0 0);
   array sigma[2, 2];
   array S[2,2] (1 0, 0 1);

   parms sigma;
   prior sigma ~ iwish(3, S);
   rho = sigma[1,2]/sqrt(sigma[1,1]*sigma[2,2]);
   model x ~ mvn(mu, sigma) monitor=(x);          ! outputs X_mis
run;
```

This is virtually identical to the program for fitting a bivariate normal data without missing values.

Procedure Outputs

Number of Observations Read	12
Number of Observations Used	12

Missing Data Information Table			
Variable	Number of Missing Obs	Observation Indices	Sampling Method
x1	4	9 10 11 12	Direct
x2	4	5 6 7 8	Direct

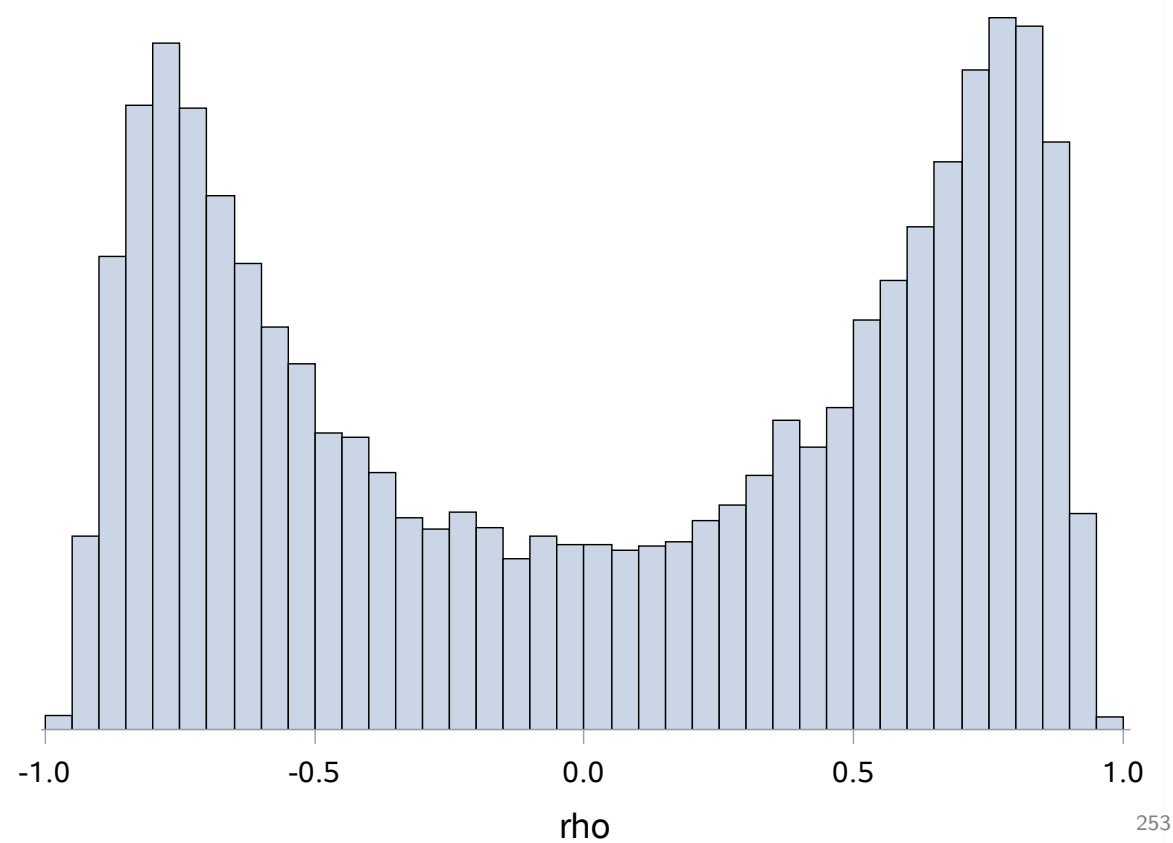
Posterior Estimates

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
rho	20000	0.0270	0.6272	-0.6197	0.0772	0.6519
x2_5	20000	0.0473	1.8898	-1.3000	0.0657	1.3932
x2_6	20000	0.0700	1.8839	-1.2648	0.0891	1.4010
x2_7	20000	-0.0652	1.9042	-1.4005	-0.0659	1.2936
x2_8	20000	-0.0746	1.9147	-1.4345	-0.0825	1.2706
x1_9	20000	0.0575	1.8807	-1.2914	0.0834	1.3871
x1_10	20000	0.0606	1.8876	-1.2808	0.0785	1.3945
x1_11	20000	-0.0497	1.8942	-1.4079	-0.0661	1.2815
x1_12	20000	-0.0456	1.8839	-1.3938	-0.0762	1.2909

Posterior Interval Estimates

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
rho	0.050	-0.8845	0.8870	-0.8796	0.8895
x2_5	0.050	-3.5461	3.6009	-3.5456	3.6009
x2_6	0.050	-3.5444	3.6455	-3.5311	3.6503
x2_7	0.050	-3.7211	3.5754	-3.5998	3.6847
x2_8	0.050	-3.7148	3.5835	-3.6730	3.6130
x1_9	0.050	-3.5345	3.6513	-3.5372	3.6432
x1_10	0.050	-3.5735	3.5786	-3.4702	3.6568
x1_11	0.050	-3.6167	3.5969	-3.7271	3.4786
x1_12	0.050	-3.5976	3.5840	-3.5339	3.6202

Estimate of the Correlation Parameter



Outline

Example Data Set

- The data are based on a double-blind antidepressant clinical trial originally reported by Goldstein et al (2004).
- The Drug Information Association (DIA) working group on missing data have made this data set available at www.missingdata.org.uk.
- To avoid implications for marketed drugs, all patients who took active medication are grouped into a single `DRUG` group and only a subset of the original trial patients are included.
- There are 171 subjects in the data set, 88 in the control arm, and 83 in the active arm.

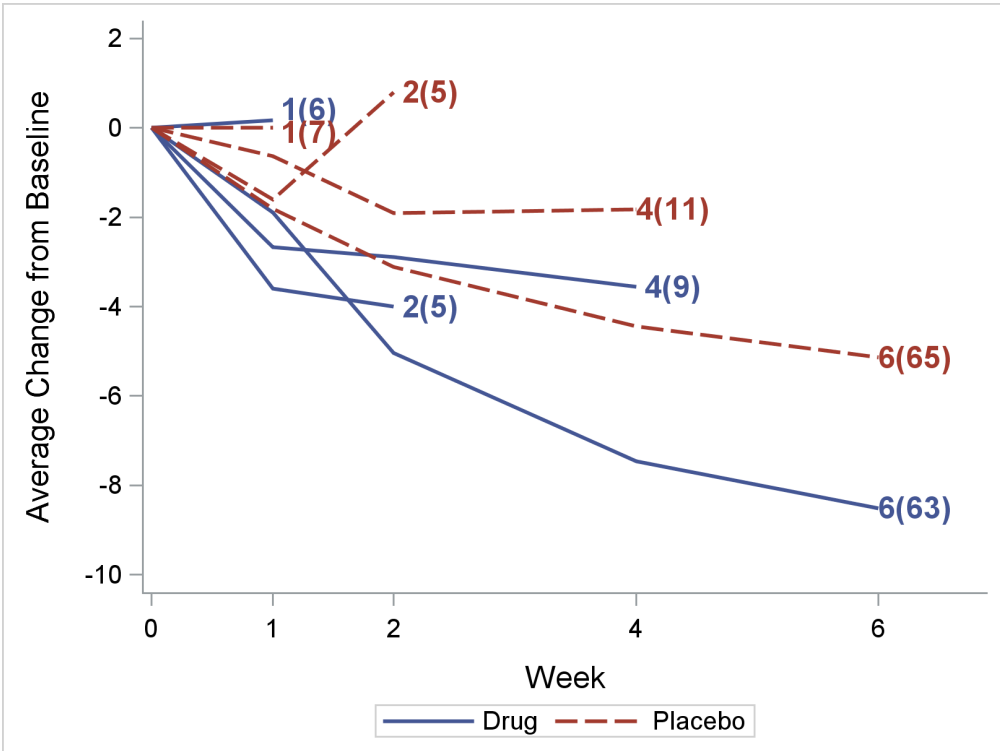
Variables in the Data Set

- `patient`: patient ID
- `baseval`: baseline assessment on the Hamilton 17-item rating scale for depression ($HAMD_{17}$, Hamilton 1960).
- `change1`–`change4`: change in $HAMD_{17}$ at weeks 1, 2, 4, and 6.
- `r1`–`r4`: missing data indicator for each of the `change` variables.
- `therapy`: treatment (`DRUG` vs `PLACEBO`)
- `poolinv`: blocking information (Groups formed by pooling investigator).
- `last`: week index to last non-missing `change` value. Patient's last visit week.
- `wkMax`: maximum number of weeks to be included in the analysis.

The first few observations of the `selection` data set:

```
data selection;
input PATIENT baseval change1-change4 r1-r4 THERAPY $ POOLINV $ last wkMax;
datalines;
 1503   32  -11  -12  -13  -15  0  0  0  0  DRUG      006   4  4
 1507   14   -3    0   -5   -9  0  0  0  0  PLACEBO   006   4  4
...
```


Average Mean Changes of $HAMD_{17}$ by Withdrawl Pattern



Data Characteristics

- Dropout probabilities appear to be correlated with the observed level of improvement (change in score).
- Patients failing to see improvement (flat or up-swinging lines), are more likely to withdraw.
- The probability of withdrawal *could* also depend on how they felt at the first unobserved visit - *the MNAR part of the model*.
- Fit a selection model:

$$f(\texttt{change}|\texttt{x}, \theta) \cdot f(\texttt{r}|\texttt{change}, \phi)$$

Outcome Model

For every subject i, $\texttt{change}_i = \{\texttt{change}_{j\,i}\}$ is modeled using a MVN($\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}$), where $j = \{1, 2, 3, 4\}$ is the week index.

The mean variables, $\boldsymbol{\mu}_i = (\mu_{1\,i}, \mu_{2\,i}, \mu_{3\,i}, \mu_{4\,i})$, are modeled via:

$$\mu_{j\,i} = m_{kj} + \beta_j \cdot (\texttt{baseval-18}) + \gamma_l$$

where $k = \{1, 2\}$ indexes the treatment, l indexes pooling investigator.

The following prior distributions are used in the analysis:

$$\begin{aligned} \pi(m_{kj}, \beta_j, \gamma_l) &\propto 1 \\ \boldsymbol{\Sigma} &\sim \text{iWishart}(4, \mathbf{I}) \end{aligned}$$

The Selection Model

The selection model (Diggle-Kenward model) includes the previous and current (possibly missing) response variables for each week:

$$\begin{aligned} \mathtt{r}_{kj\,i} &\sim \text{binary}(q_{kj\,i}) \\ q_{kj\,i} &= \text{logistic}(\phi_{k1} + \phi_{k2} \cdot \text{change}_{(j-1)\,i} + \phi_{3k} \cdot \text{change}_{j\,i}) \end{aligned}$$

The parameters $\phi_{k\cdot}$ account for treatment effect in separate regression models. Flat prior is used:

$$\pi(\phi_{k\cdot}) \quad \propto 1$$


```
proc mcmc data=selection nmc=20000 seed=176 outpost=seleout;
   array Change[4] Change1-Change4;                    ! response
   array mu[4];                                        ! μ_i
   array Sigma[4,4];                                   ! Σ
   array S[4,4] (1 0 0 0, 0 1 0 0, 0 0 1 0, 0 0 0 1);! S = I
   array beta[4] ;                                     ! β_j
   array M[2,4] m1-m8;                                 ! m_kj
   array phi[2,3] phi1-phi6;                           ! φ_k.

   parms beta: 0 ;
   parms m1-m8 0;
   parms phi1-phi6 0;
   parms Sigma ;
   prior beta: m1-m8 phi: ~ general(0);                ! π(m_kj, β_j, φ_k) ∝ 1
   prior Sigma ~ iwish(4, S);                          ! π(Σ) = iWishart(4, S)

   /* outcome model */
   random gamma ~ general(0) subject=poolinv zero=first init=0;   ! π(γ_l) ∝ 1
   do j=1 to 4;
      if therapy eq "DRUG" then do;
         mu[j] = m[1,j] + gamma + beta[j]*(baseval-18); ! μ_{k = DRUG}j
      end; else do;
         mu[j] = m[2,j] + gamma + beta[j]*(baseval-18); ! μ_{k = PLACEBO}j
      end;
   end;
   model Change ~ mvn(mu, Sigma);                      ! likelihood
```

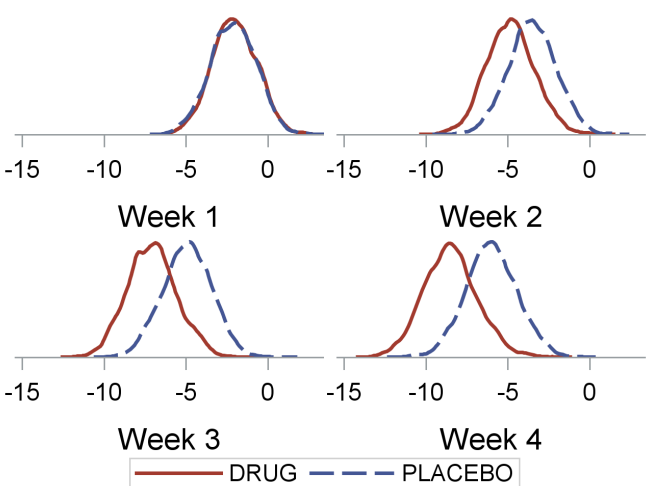
MCMC Code for the Selection Model

```
   /* selection mechanism */
   array r[4] r1-r4;                                  ! missing data indicator
   llike = 0;
   do j = 2[3] to wkMax
      if therapy eq "DRUG" then do;
         mn = phi[1,1] + phi[1,2] * change[j-1] + phi[1,3] * change[j];
         q = logistic(mn);                            ! q_{k=DRUG}j
      end; else do;
         mn = phi[2,1] + phi[2,2] * change[j-1] + phi[2,3] * change[j];
         q = logistic(mn);                            ! q_{k=PLACEBO}j
      end;
      llike = llike + lpdfbern(r[i], q);              ! accumulates binary
                                                      ! likelihood over weeks
   end;
   model r2 r3 r4 ~ general(llike);                   ! declares joint likelihood
run;
```

[3] Variable `change1` doesn't contain any missing values, making `r1` irrelevant to the analysis.

Outcome Model Estimates

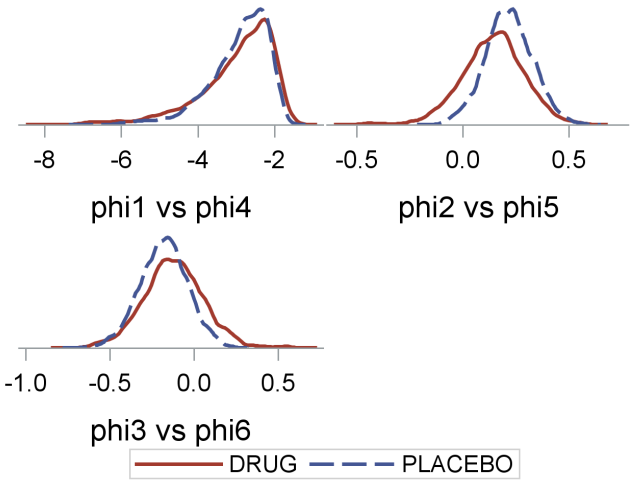
Comparison of posterior distributions of $m_{\text{drug},j}$ and $m_{\text{placebo},j}$ over the weeks:



- The treatment difference at week 1 is negligible.
- The difference becomes larger as the trial progresses, with the predicted score change for the DRUG group declining at a faster pace. The difference (mean difference is -2.42) is largest at the end of the trial.

Selection Model Estimates, When All are Estimated

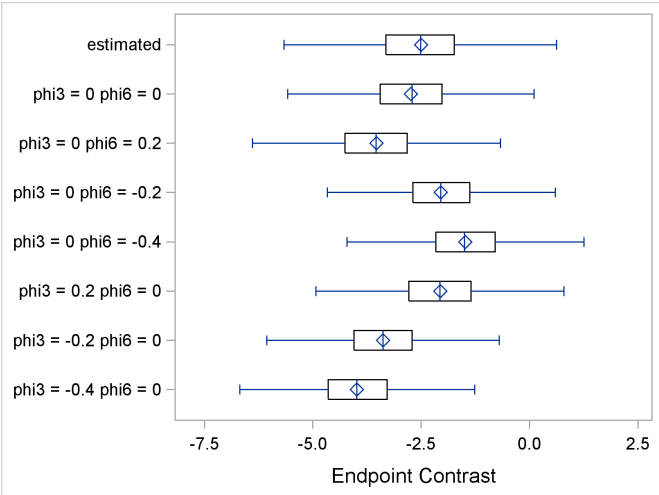
Posterior distributions of $\phi_{k\cdot}$, which model the change in the probability of dropouts given the score changes in the last and the current, potentially missing, week:



- $\phi_{\text{drug},2}$ (`phi2`) and $\phi_{\text{placebo},2}$ (`phi5`) are positive, suggesting that as the patient felt *worse* (increase in HAMD_{17} score) in their previous visit, they were *more* likely to dropout.
- $\phi_{\text{drug},2}$ (`phi3`) $\phi_{\text{placebo},2}$ (`phi6`) are negative, suggesting that patients were *less* likely to withdraw from the trial had they felt *worse* in the current week.

Sensitivity Analysis Fixing MNAR Parameter Values

The parameters in this complete model are poorly estimated. An idea is to fix the regression on the potentially unobserved values (`phi3` and `phi6`) and observe sensitivity to changing these. The estimated model (1st boxplot) produces similar point estimates (but larger s.d.) to the MAR model (2nd).



- when `phi3` $<$ `phi6`, boxplots shift to the left (3rd, 7th, and 8th). `DRUG` patients were more likely to drop out if they felt improvement in the current week. This results in stronger estimated treatment effect as the estimate is *corrected* for these missed patients.
- when `phi3` $>$ `phi6`, boxplots shift to the right (4th, 5th, and 6th), resulting in weaker treatment effect estimates.

Outline

Piecewise Exponential Model

Let $\{(t_i, \mathbf{x}_i, \delta_i), i = 1, 2, \ldots, n\}$ be the observed data. Let $a_0 = 0 < a_1 < \ldots < a_{J-1} < a_J = \infty$ be a partition of the time axis. The hazard for subject i is

$$h(t|\mathbf{x}_i; \boldsymbol{\theta}) = h_0(t) \exp(\boldsymbol{\beta}' \mathbf{x}_i)$$

where

$$h_0(t) = \lambda_j \qquad a_{j-1} \leq t < a_j \qquad (j = 1, \ldots, J)$$

The hazard for subject i in the jth time interval is

$$h(t) = \lambda_j \exp(\boldsymbol{\beta}' \mathbf{x}_i) \qquad a_{j-1} < t < a_j$$

Piecewise Exponential Model

From the hazard function, first define the baseline cumulative hazard function:

$$H_0(t) = \sum_{j=1}^{J} \lambda_j \Delta_j(t)$$

where

$$\Delta_j(t) = \begin{cases} 0 & t < a_{j-1} \\ t - a_{j-1} & a_{j-1} \leq t < a_j \\ a_j - a_{j-1} & t \geq a_j \end{cases}$$

Piecewise Exponential Model

The log likelihood is:

$$\begin{aligned} l(\boldsymbol{\lambda}, \boldsymbol{\beta}) &= \sum_{i=1}^{n} \delta_i \left[\sum_{j=1}^{J} I(a_{j-1} \leq t_i < a_j) \log \lambda_j + \boldsymbol{\beta}' \mathbf{x}_i \right] \\ &\quad - \sum_{i=1}^{n} \left[\sum_{j=1}^{J} \Delta_j(t_i) \lambda_j \right] \exp(\boldsymbol{\beta}' \mathbf{x}_i) \end{aligned}$$

where δ_i is the event status:

$$\delta_i = \begin{cases} 0 & \text{if } t_i \text{ is a censored time} \\ 1 & \text{if } t_i \text{ is an event time} \end{cases}$$

This model has two parameter vectors: $\boldsymbol{\lambda}$ and $\boldsymbol{\beta}$.

Fitting Piecewise Exponential Models Using PROC MCMC

- Road map (what needs to be done)
  - Reformulate the likelihood to a Poisson likelihood, which enables us to treat hazards as random effects
  - Manipulate the data
  - Fit using PROC MCMC
  - Extend to frailty model

Fitting Piecewise Exponential Models Using PROC MCMC

Recall the hazard function

$$h(t|\mathbf{x}_i;\boldsymbol{\theta}) = h_0(t)\exp(\boldsymbol{\beta}'\mathbf{x}_i)$$

Define $N_i(t)$ to be the number of observed failures of the ith subject up to time t, then the hazard function is a special case of a *multiplicative intensity model* (Clayton, 1991, Biometrics, 467-485). And the intensity process for $N_i(t)$ becomes

$$I_i(t) = Y_i(t)h_0(t)\exp(\beta'\mathbf{x}_i)$$

where

$$Y_i(t) = \begin{cases} 1 & \text{if the subject is observed at time } t \\ 0 & \text{o.w.} \end{cases}$$

Under *noninformative censoring*, the corresponding likelihood is proportional to

$$\prod_{i=1}^{n}\left[\prod_{t\geq 0} I_i(t)\right]^{dN_i(t)} \exp\left[-\int_{t\geq 0} I_i(t)dt\right]$$

where $dN_i(t)$ is the increment of $N_i(t)$ over the small time interval $[t, t+dt)$:

$$dN_i(t) = \begin{cases} 1 & \text{if the subject } i \text{ fails in the time interval} \\ 0 & \text{o.w.} \end{cases}$$

Poisson Process as the Likelihood Function

This is a Poisson kernel with the random variable being the increments of dN_i and the means $I_i(t)dt$

$$dN_i(t) \sim \text{Poisson}(I_i(t)dt)$$

where

$$I_i(t)dt = Y_i(t)\exp(\beta'\mathbf{x})h_0(t)$$

and

$$h_0(t) = \int_0^t h_0(u)du.$$

The integral is the increment in the integrated baseline hazard function that occurs during the time interval $[t, t+dt)$.

Alternative Approach

The alternative formulation of the piecewise exponential model

$$dN_i(t) \sim \text{Poisson}(Y_i(t)\exp(\beta'\mathbf{x})h_0(t))$$

makes it a random-effects model, with each hazard rate, $h_0(t)$ being a random effect.

You need to manipulate the data and create $Y_i(t)$ and $dN_i(t)$ for each interval.

Piecewise Exponential Model

Consider a randomized trial of 40 rats exposed to carcinogen:

- Drug X and Placebo are the treatment groups.
- Event of interest is death.
- Response is time until death.
- What are the effects of treatment and gender on survival?

Piecewise Exponential Model

A subset of the data:

```
proc format;
   value Rx 1='X' 0='Placebo';
data Exposed;
  input Days  Status Trt Gender $ @@;
  format Trt Rx.;
  datalines;
179   1   1   F   378   0   1   M
256   1   1   F   355   1   1   M
262   1   1   M   319   1   1   M
256   1   1   F   256   1   1   M
...
268   0   0   M   209   1   0   F
;
```


Piecewise Exponential Model

The following regression model and prior distributions are used in the analysis:

$$\begin{aligned} \beta' \mathbf{x}_i &= \beta_1 \text{Trt} + \beta_2 \text{Gender} \\ \beta_1, \beta_2 &\sim \text{normal}(0, \text{var} = 1e6) \\ h_0(t) &\sim \text{gamma}(\text{shape} = 0.5, \text{iscale} = 0.5) \end{aligned}$$

where $t = 1 \cdots 8$ (time intervals).

A Little Problem

Both `Trt` and `Gender` are character variables and PROC MCMC does not support a CLASS statement.

PROC TRANSREG to the rescue:

```
proc transreg data=exposed design;
   model class(trt gender / zero=first);
   id days status id;
   output out=exposed_d(drop=_: Int:);
   run;
```

design : specifies design matrix coding
class : expands the variables to "dummy" variables
zero : controls reference level (`first` sets 0 to the first sorted category)

```
zero="X" "F"
```

ID : includes additional variables to the OUT= data set
OUTPUT : creates a new data set

The New Data Set

Obs	Days	Status	Trt	Gender
1	179	1	X	F
2	378	0	X	M
3	256	1	X	F
4	355	1	X	M
5	262	1	X	M
6	319	1	X	M
7	256	1	X	F
8	256	1	X	M
9	255	1	X	M
10	171	1	X	F

$\Rightarrow$

		Gender		
Obs	TrtX	M	Days	Status
1	1	0	179	1
2	1	1	378	0
3	1	0	256	1
4	1	1	355	1
5	1	1	262	1
6	1	1	319	1
7	1	0	256	1
8	1	1	256	1
9	1	1	255	1
10	1	0	171	1

Automatically created macro variable `&_trgind` contains the list of independent variables created:

```
%put &_trgind;
TrtX GenderM
```

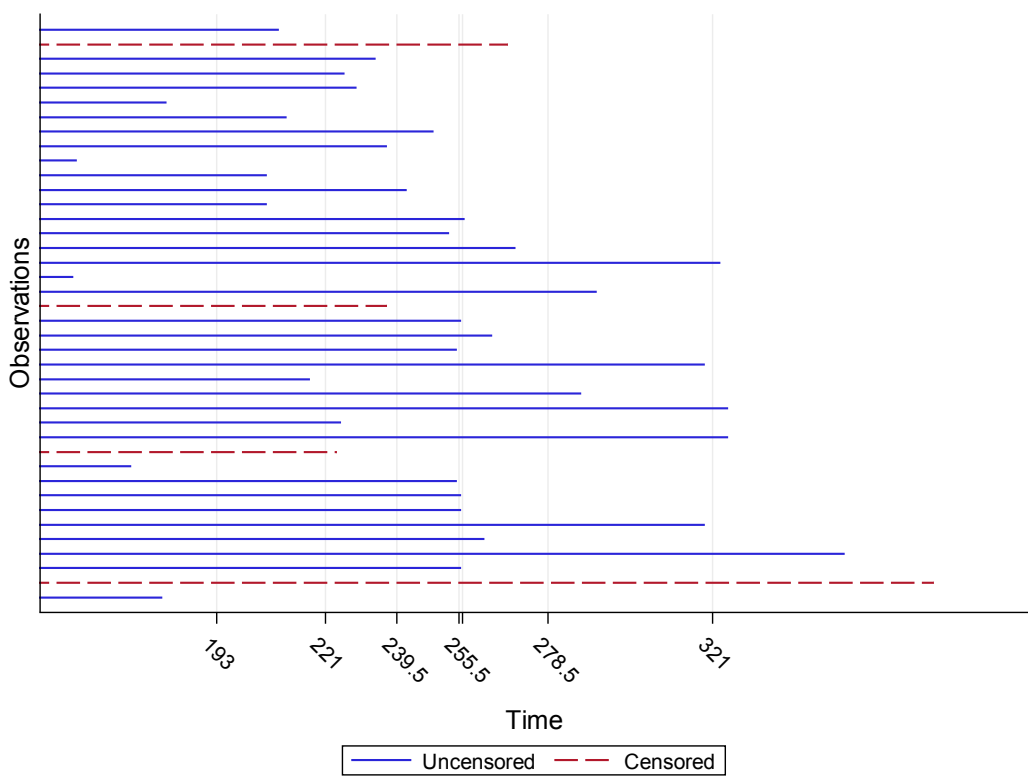
Partition of the Time Axis

You can find a partition along the time axis using PROC UNIVARIATE, placing roughly the same number of event times in each interval:

```
%let npar_e = 7;
%let inc = %sysevalf((1/&npar_e) * 100);
proc univariate data=exposed_d(where=(status=1)) pctldef=4;
   var days;
   output out=interval pctlpre=P_ pctlpts= 0 to 100 by &inc;
   run;
```

Alternatively, we use the same partition as PROC PHREG:

```
data partition;
   input int_1-int_9;
datalines;
 0  193 221 239.5 255.5 256.5 278.5 321 1000
;
```

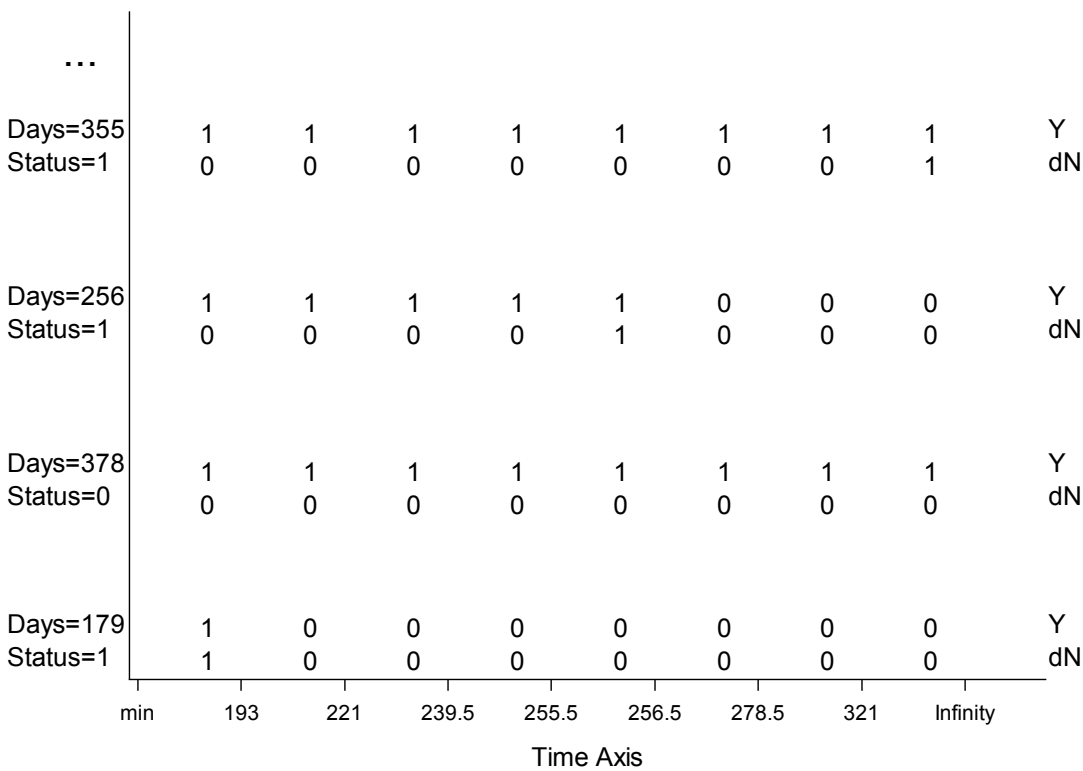
Manipulate the Data

The next step is to create $Y_i(t)$ and $dN_i(t)$ using `Days` and `Status`:

$$Y_i(t) = \begin{cases} 1 & \text{if the subject is observed at time } t \\ 0 & \text{o.w.} \end{cases}$$

$$dN_i(t) = \begin{cases} 1 & \text{if the subject } i \text{ failes in the time interval} \\ 0 & \text{o.w.} \end{cases}$$

Create $Y_i(t)$ and $dN_i(t)$



Modify the Data

The following statements calculate $Y_i(t)$ for each observation i, at every time point t in the Partition data set. The statements also find the observed failure time interval, $dN_i(t)$, for each observation:

```
%let n = 8;
data _a;
   set exposed_d;
   if _n_ eq 1 then set partition;
   array int[*] int_:;
   array Y[&n];
   array dN[&n];
   do k = 1 to &n;
      Y[k] = (days - int[k] + 0.001 >= 0);
      dN[k] = Y[k] * ( int[k+1] - days - 0.001 >= 0) * status;
      end;
   output;
   drop int_: k;
run;
```


First few observations of the new data set:

Obs	ID	Time	Status	DT	dY1	dY2	dY3	dY4	dY5	dY6	dY7	dY8	dN1	dN2	dN3	dN4	dN5	dN6	dN7	dN8
1	5	46.23	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
2	5	46.23	0	1	0	2	1	1	1	1	1	1	1	0	0	0	0	0	0	0
3	14	42.50	0	0	1	3	1	1	1	1	1	1	1	0	0	0	0	0	0	0
4	14	31.30	1	0	0	4	1	1	1	1	1	1	0	0	0	0	0	0	1	0
5	16	42.27	0	0	1	5	1	1	1	1	1	1	1	0	0	0	0	0	0	0
6	16	42.27	0	0	0	6	1	1	1	1	1	1	1	0	0	0	0	0	0	0
...																				

This being the Partition data set:

0	193	221	239.5	255.5	256.5	278.5	321	1000

Input Data Set

- Each observation in the _a data set has 8 Y and 8 dN, meaning that you would need eight MODEL statements in a PROC MCMC call, each for a Poisson likelihood.

```
model dN1 ~ poisson(Y1 * exp(beta * x) * h1);
model dN2 ~ poisson(Y2 * exp(beta * x) * h2);
...
model dN8 ~ poisson(Y8 * exp(beta * x) * h8);
```

- Alternatively, you can expand _a, put one Y and one dN in every observation, and fit the data using a single MODEL statement in PROC MCMC. This enables you to treat the hazards ($h_0(t)$) as random-effects and use the RANDOM statement.

The following statements expand the data set _a and save the results in the data set _b:

```
data _b;
   set _a;
   array y[*] y:;
   array dn[*] dn:;
   do i = 1 to (dim(y));
      y_val       = y[i];
      dn_val      = dn[i];
      int_index   = i;
      output;
      end;
   keep y_: dn_: &_trgind int_index id;
run;

data _b;
   set _b;
   rename y_val=Y dn_val=dN;
run;
```

The data set _b now contains 320 observations. The int_index variable is an index variable that indicates interval membership of each observation.

Obs	TrtX	Gender M	id	Y	dN	int_ index
1	1	0	1	1	1	1
2	1	0	1	0	0	2
3	1	0	1	0	0	3
4	1	0	1	0	0	4
5	1	0	1	0	0	5
6	1	0	1	0	0	6
7	1	0	1	0	0	7
8	1	0	1	0	0	8
9	1	1	2	1	0	1
10	1	1	2	1	0	2
...						

Further Clean Up the Data

Recall the likelihood is:

$$dN_i(t) \sim \text{Poisson}(Y_i(t)\exp(\beta'\mathbf{x})h_0(t))$$

where $Y_i(t)$ does not contribute to the likelihood calculation when it takes a value of 0, you can remove these observations.

```
data inputdata;
   set _b;
   if Y > 0;
run;
```

This steps reduces the size of the input data set (to 174 observations) and shortens the run time of PROC MCMC.

Fitting Piecewise Exponential Model Using PROC MCMC

The following statements fit a piecewise exponential model in PROC MCMC:

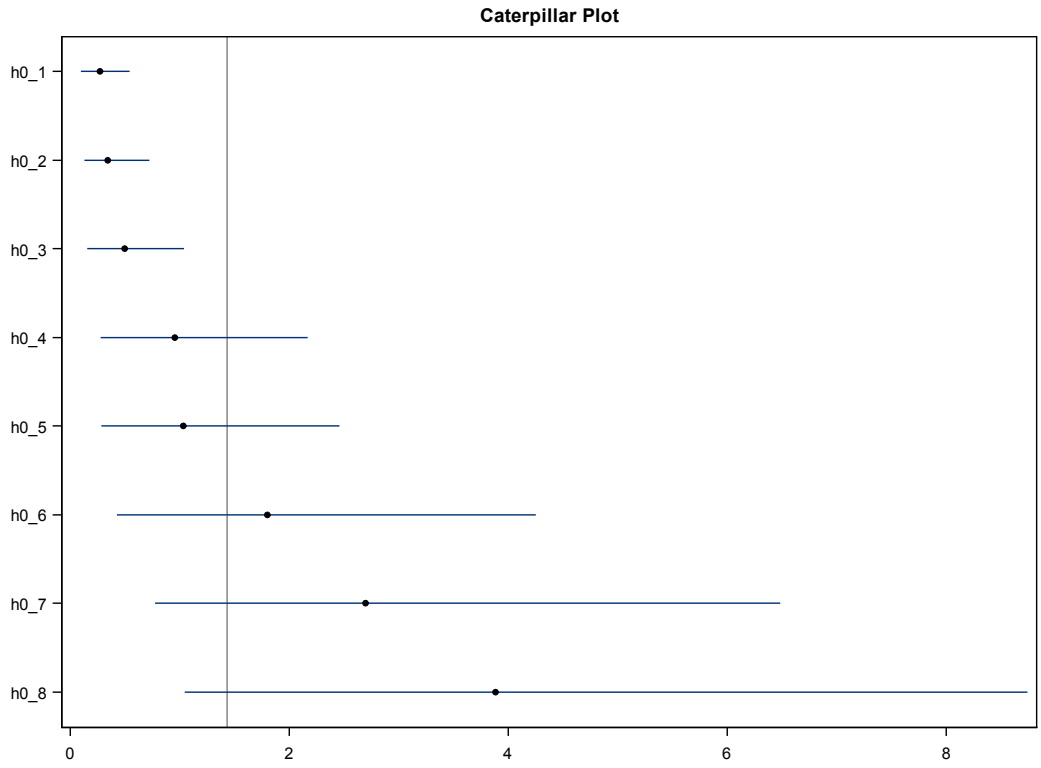
```
proc mcmc data=inputdata nmc=10000 outpost=postout seed=12351
   stats=summary diag=ess;
   parms beta1-beta2 0;
   prior beta: ~ normal(0, var = 1e6);
   random h0 ~ gamma(0.5, iscale = 0.5) subject=int_index;
   bZ = beta1*trtx + beta2*genderM;
   idt = exp(bz) * h0;
   model dN ~ poisson(idt);
   run;
```

Note that the $Y_i(t)$ term is omitted in the assignment statement for the symbol `idt` because `Y` $= 1$ in all observations.

Posterior Estimates

Posterior Summaries						
				Percentiles		
Parameter	N	Mean	Standard Deviation	25%	50%	75%
beta1	10000	-0.5659	0.3288	-0.7869	-0.5666	-0.3514
beta2	10000	-1.5919	0.3520	-1.8310	-1.5916	-1.3586

Hazards Estimates



Frailty Model

Now suppose you want to include patient-level information and fit a frailty model to the exposed data set.

$$
\begin{aligned}
\beta' \mathbf{x}_i &= \beta_1 \text{Trt} + \beta_2 \text{Gender} + u_{id} \\
u_{id} &\sim \text{normal}(0, \text{var} = \sigma^2) \\
\sigma^2 &\sim \text{igamma}(\text{shape} = 0.01, \text{scale} = 0.01) \\
\beta_1, \beta_2 &\sim \text{normal}(0, \text{var} = 1e6) \\
h_0(t) &\sim \text{gamma}(\text{shape} = 0.5, \text{iscale} = 0.5) \\
dN_i(t) &\sim \text{Poisson}(Y_i(t) \exp(\beta' \mathbf{x}) h_0(t))
\end{aligned}
$$

where $t = 1 \cdots 8$ and `id` indexes patient.

The actual coding in PROC MCMC of a piecewise exponential frailty model is rather straightforward:

```
proc mcmc data=inputdata nmc=10000 outpost=postout seed=12351
     stats=summary diag=none;
   parms beta1-beta2 0 s2;
   prior beta: ~ normal(0, var = 1e6);
   prior s2 ~ igamma(0.01, scale=0.01);
   random h0 ~ gamma(0.01, iscale = 0.01) subject=int_index;
   random u ~ normal(0, var=s2) subject=id;
   bZ = beta1*trtx + beta2*genderM + u;
   idt = exp(bZ) * lambda;
   model dN ~ poisson(idt);
   run;
```

And you are done!

Learning Objectives

Attendees will

- understand basic concepts and computational methods of Bayesian statistics
- be able to deal with some practical issues that arise from Bayesian analysis
- be able to program using SAS/STAT procedures with Bayesian capabilities to implement various Bayesian models.