In many Bayesian applications, one cannot analytically solve for the full posterior density as we did in our simple example. In that case, we might be able to generate a random sample from the posterior to draw inferences. In fact sometimes even when the posterior density is known, it can be easier to draw inferences based on a sample from the posterior.

In general, Bayesians use $g(\mu)$ to denote the function of interest. The function of interest may be quite simple as in our initial example $g(\mu)=\mu$ or it could be more complex, $g(\mu)=\exp(\mu)/\mu^2$. In an application where $\mu$ is replaced by a vector of population parameters, the function interest may involve any or all of the population parameters. Using numerical techniques greatly simplifies our work for more complex problems.

Assuming a quadratic loss, our point estimator for $g(\mu)$ is the posterior mean. Thus, we seek to determine:

$$E[g(\mu)] = \int g(\mu) p(\mu|y) d\mu$$

When no solution exists for this integral, we resort to numerical integration, typically by Monte Carlo methods. Assuming we can sample $N$ points from $p(\mu|y)$, label each as $\mu_i$, then we can approximate the mean using:

$$\bar{g} = \frac{1}{N} \sum_{i=1}^{N} g(\mu_i) \rightarrow \int g(\mu) p(\mu | y) d\mu$$
N is called the number of Monte Carlo repetitions and can be increased to obtain more precise estimates of the posterior mean. In addition, the posterior variance, percentiles and marginal densities can be approximated using a random sample from the posterior distribution. For example the variance of \( g(\mu) \) can be approximated arbitrarily well by:

\[
\text{var}(g(\mu)) = \frac{1}{N} \sum_{i=1}^{N} (g(\mu_i) - \bar{g})^2 = \frac{1}{N} \sum_{i=1}^{N} g(\mu_i)^2 - \bar{g}^2
\]

Notice that this variance measures the dispersion of the function of interest itself. We can also calculate a measure for the precision of our numerical estimator for the mean. If we sample directly from the posterior, the numerical standard error is \( \sqrt{\frac{\text{var}(g(\mu))}{N}} \). Notice that the numerical error can be made arbitrarily small through increasing the number of repetitions. For more complex algorithms which do not sample directly from the posterior this formula is not valid! In addition, this formula only measures the numerical error in approximating the mean. The numerical error may be larger if the goal is to approximate other moments or tail areas of the density.

Numerical algorithms in Bayesian statistics typically involved generating a sample from the posterior (or a distribution similar to the posterior) and computing moments for functions of interest using this sample. Notice that there is no optimization in a typical Bayesian routine, unlike frequentist algorithms which generally focus on numerical optimization.

At this point, you are ready to complete problem set #1. This exercise asks you to construct a sampler for the normal distribution and evaluate functions of interest. After completing problem set #1, try answering questions A-D on the basis of samples of size
1,000 and 100,000 from the posterior density. Are you able to obtain the same answer that we calculated numerically?

At this point you should realize that generating a sample from the posterior density would allow us to draw inferences to problems where analytic calculations are impossible. Even if you cannot write a closed form expression for the posterior, inference is often possible using numerical algorithms. Geweke’s (1998) article in the Handbook of Computational statistics summarizes many methods of generating these samples. The most common methods are discussed in section II of his notes as well.

**The Gibbs Sampler**

In most applications involving several parameters it is impossible to sample directly from the posterior density. This problem greatly limited the number of empirical applications of Bayesian statistics prior to around 1990. For most real problems, posterior moments were impossible to evaluate analytically and the Bayesian approach was either extremely difficult or impossible to implement numerically as well. Recent advances in sampling theory have revolutionized the field and made Bayesian applications practical for an overwhelming majority of statistical problems, even in many applications where frequentist methods are not possible.

One key advance in sampling theory is the **Gibbs sampler**. This sampler is applicable when one cannot generate a sample from the joint posterior, but can sample from conditional densities for subsets of the parameter space. The **Gibbs sampler** allows us to generate the samples needed to draw inferences for a host of standard econometric problems. The Bayesian approach to linear regression, tobit, logit, probit, SUR, and numerous other models relies on the Gibbs.
To demonstrate the Gibb’s sampler and move into inference for a multivariate parameter space, turn to problem set #2. In this problem, you are asked to consider a multivariate normal distribution. The following facts are useful.

The Bivariate Normal:

Consider $Z = [X, Y]' \sim N(\mu, \Sigma)$,

with $\mu = [\mu_X, \mu_Y]$, and $\Sigma = \begin{bmatrix} \sigma_X^2 & \rho \sigma_X \sigma_Y \\ \rho \sigma_X \sigma_Y & \sigma_Y^2 \end{bmatrix}$, $-1 < \rho < 1$.

A bit of calculus would show the marginal and conditional densities are:

$$
X \sim N(\mu_X, \sigma_X^2) \\
Y \sim N(\mu_Y, \sigma_Y^2) \\
Y | X = x \sim N(\mu_{Y|x}, \sigma_{Y|x}^2) \\
$$

where $\mu_{Y|x} = \mu_Y + \rho \frac{\sigma_Y}{\sigma_X} (x - \mu_X)$.

$$
\sigma_{Y|x} = \sigma_Y^2 (1 - \rho^2) \\
X | Y = y \sim N(\mu_{X|y}, \sigma_{X|y}^2) \\
$$

where $\mu_{X|y} = \mu_X + \rho \frac{\sigma_X}{\sigma_Y} (y - \mu_Y)$.

$$
\sigma_{X|y} = \sigma_X^2 (1 - \rho^2) \\
$$

To use the Gibb’s sampler to generate a sample from this density, the following steps are needed:

1. Pick an initial value for X, say $X^{(0)}$.
2. Sample $Y^{(1)}$ from $Y | X = X^{(0)}$.
3. Sample $X^{(1)}$ from $X | Y = Y^{(1)}$.
4. Sample $Y^{(2)}$ from $Y | X = X^{(1)}$.
5. Sample $X^{(2)}$ from $X | Y = Y^{(2)}$.
6. Iterate to generate a sample of size N.
Assuming the conditions summarized in Geweke's notes are satisfied, this algorithm will generate a sample that converges to the joint posterior. The key condition is that the probability of sampling any point in the posterior domain must be non-zero for the sampler.

To eliminate dependence on the initial value (which was chosen arbitrarily), we typically drop the first points generated by the algorithm. A rule of thumb might be to drop the first 1000 points or so. Another possibility is to only use every 100th (X,Y) combination. This minimizes the serial correlation in the sample.