Stat 544 Spring 2008
Homeworks

In what follows, unless otherwise indicated, we will use the GCS&R notations/parameterizations for distributions (see their Appendix A that begins on page 571). If there is a conflict between GCS&R and WinBUGS notations and we wish to refer to the latter, we'll preface the distribution's name with the modifier "WinBUGS-" (so, for example, $\mathrm{N}(\mu, \sigma^2)$ is, according to the GCS&R convention, the normal distribution with mean μ and variance σ^2, while WinBUGS-$\mathrm{N}(\mu, \tau^2)$ is the normal distribution with mean μ and variance $1/\tau^2$ (i.e. "precision" τ^2)).

Homework #1 (Due February 1, 2008 5PM at the TA's Office in Wilson if not turned in at class or during her office hours in Pearson 0113)

Read the first 2 Chapters of GCS&R.

1. Do problems 2.1, 2.2, 2.5, 2.8, 2.12, 2.21, 2.22 from GCS&R

2. Consider a Poisson model like the Binomial one discussed in class. That is, suppose that conditioned on λ, Y_1 and Y_2 are independent Poisson(λ) random variables.

   (a) First consider the possibility of using a Gamma(α, β) prior for λ. Work out analytically the posterior distribution of $\lambda | Y_1 = y_1$ and (possibly only up to a constant multiplier) the posterior predictive distribution of $Y_2 | Y_1 = y_1$. For the choices $(\alpha, \beta) = (1, 1)$ and $(\alpha, \beta) = (10, 10)$ plot both the prior density for λ and the posterior density for $\lambda | Y_1 = 0$. For the choices $(\alpha, \beta) = (1, 1)$ and $(\alpha, \beta) = (10, 10)$ plot in histogram form (possibly only up to a multiplicative constant) prior probabilities for $Y_2 = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$ and corresponding posterior probabilities for $Y_2 | Y_1 = 0$.

   (b) Now consider the possibility of using a Uniform$(0, 10)$ prior for λ. Work out analytically the posterior distribution of $\lambda | Y_1 = 0$ and plot both the prior density for λ and the posterior density for $\lambda | Y_1 = 0$.

   (c) Use `WinBUGS` and get approximations for the posterior distribution of $\lambda | Y_1 = 3$ and the posterior predictive distribution of $Y_2 | Y_1 = 3$ based on Gamma(α, β) priors for λ for the choices $(\alpha, \beta) = (1, 1)$ and $(\alpha, \beta) = (10, 10)$ and based on the Uniform$(0, 10)$ prior. (Use the "density" and "stats" functions on the "Sample Monitor" tool to summarize these approximations. You can easily/quickly generate 100,000 or so iterations in this simple problem.)

   (d) Redo part (c) using `WinBUGS` for the case where $Y_1 = 7$ is observed (instead of $Y_1 = 3$).

In parts (c) and (d) above you will probably want to use some `WinBUGS` code like:

```
model {
Y1~dpois(lambda)
Y2~dpois(lambda)
lambda~dgamma(a,b)
}

list(a=1,b=1,Y1=3)

model {
Y1~dpois(lambda)
Y2~dpois(lambda)
lambda~dunif(a,b)
}

list(a=0,b=10,Y1=3)
```

Homework #2 (Due February 15, 2008 5PM)

1. Consider the following model. Given parameters $\lambda_1, ..., \lambda_N$ variables $X_1, ..., .X_N$ are independent Poisson variables, X_i ~Poisson(λ_i). M is a parameter taking values in $\{1, 2, ..., N\}$ and if $i \leq M$, $\lambda_i = \mu_1$, while if $i > M$, $\lambda_i = \mu_2$. (M is the number of Poisson means that are the same as that of the first observation.) With parameter vector $\theta = (M, \mu_1, \mu_2)$ belonging to $\Theta = \{1, 2, ..., N\} \times (0, \infty) \times (0, \infty)$ we wish to make inference about M based on $X_1, ..., .X_N$ in a Bayesian framework. As matters of notational convenience, let $S_m = \sum_{i=1}^{m} X_i$ and $T = S_N$.

   (a) If, for example, a prior distribution G on Θ is constructed by taking M uniform on $\{1, 2, ..., N\}$ independent of μ_1 exponential with mean 1, independent of μ_2 exponential with mean 1, it is possible to explicitly find the (marginal) posterior of M given that $X_1 = x_1, ..., X_N = x_N$. Don't actually bother to finish the somewhat messy calculations needed to do this, but show that this is possible (indicate clearly why appropriate integrals can be evaluated explicitly).

   (b) Suppose now that G is constructed by taking M uniform on $\{1, 2, ..., N\}$ independent of (μ_1, μ_2) with joint pdf $g(\cdot, \cdot)$ on $(0, \infty) \times (0, \infty)$. Describe in as much detail as possible a SSS based method for approximating the posterior of M, given $X_1 = x_1, ..., X_N = x_N$. (Give the necessary conditionals up to multiplicative constants, say how you're going to use them and what you'll do with any vectors you produce by simulation.)

2. A curious statistician is interested in the properties of a 5-dimensional distribution on $[0, 1]^5$ that has pdf proportional to

$$h(\boldsymbol{x}) = \exp\left[-\left((x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^2 + (x_4 - x_5)^2 + (x_5 - x_1)^2\right)\right]$$

   This person decides to simulate from this distribution. Carefully describe a (5-variate) rejection algorithm that could be used to produce realizations $\boldsymbol{x}$ from this distribution, a "Gibbs Sampler"/"Successive Substitution Sampling" algorithm, a M-H-in-Gibbs hybrid algorithm, AND a (5-variate) Metropolis-Hastings algorithm that could be used to draw samples from this distribution. For the Gibbs sampler, give a formula for the univariate pdf that must be sampled when replacing the ith entry of a current $\boldsymbol{x}$ vector. This formula will involve a 1-dimensional integral that might be evaluated numerically. For the Gibbs/M-H hybrid sampler, use univariate Metropolis-Hastings steps in your Gibbs updates. (Note, by the way, that $\boldsymbol{U} = (U_1, U_2, \ldots, U_5)$ with iid $U(0,1)$ coordinates U_i has pdf $p(\boldsymbol{u}) = 1$ on $[0,1]^5$.)

3. Here is a problem related to "scale counting." As a means of "automated" counting of a lot of discrete objects, I might weigh the lot and try to infer the number in the lot from the observed weight. Suppose that items have weights that are normally distributed with mean μ and standard deviation 1. In a calibration phase, I weigh a group of 50 of these items (the number being known from a "hand count") and observe the weight X. I subsequently weigh a large number, M, of these items and observe the weight Y. I might model $X \sim$N$(50\mu, 50)$ independent of $Y \sim$N$(M\mu, M)$. Suppose further that I use a prior distribution for (μ, M) of independence, $\mu \sim$N$(0, (1000)^2)$ and $M - 1 \sim$Poisson(100). Completely describe a simulation-based way of approximating the posterior mean of M.

4. Consider a bivariate distribution on $(0,\infty)\times(0,\infty)$ with pdf on that space proportional to

$$h(x,y) = \sin^2(x)\exp(-x-y-xy)$$

(Note that $0 \le \sin^2(x)\exp(-xy) \le 1$. Note too that given a $U(0,1)$ random observation U, you can make an Exp(β) observation as $-\frac{1}{\beta}\ln U$.) Describe in detail several algorithms that could be used to simulate from this bivariate distribution using only a stream of $U(0,1)$ random variables. In particular, describe

   (a) A rejection algorithm.

   (b) A Gibbs algorithm, where for updating x you use a univariate rejection algorithm and for updating y you use direct simulation from the conditional distribution of $Y|X = x$.

   (c) M-H within Gibbs algorithm, where for updating x you use a univariate Metropolis-Hastings step and for updating y you use direct simulation from the conditional of $Y|X = x$.

   (d) A M-H algorithm where the bivariate proposal distribution at any step is one of independent Exp(1) variables.

5. (Some more practice with `WinBUGS` in a fairly simple context.) A critical dimension is measured on $n = 5$ consecutive metal parts produced by an automated machining process. The resulting values (in units of .0001 in above an ideal value) are

$$4, 3, 3, 2, 3$$

We are going to assume that an iid N(μ, σ^2) model can be used to describe these part dimensions.

(a) Use the standard "Stat 101" t and χ^2 interval formulas and give 95% two-sided confidence limits for μ and σ. Use the formula $\bar{x} \pm ts\sqrt{1+\frac{1}{n}}$ and make 95% two-sided prediction limits for an additional value from this normal distribution.

(b) Now consider some Bayes analyses of this very small data set. There is a set of `WinBUGS` code below that can be used to analyze these data using an improper "uniform $(-\infty,\infty)$" prior for μ and a $U(0,10000)$ prior for $\tau = 1/\sigma^2$. Run 3 chains simultaneously using this code (and the last lists in the code as initializations of those chains), and monitor "history" plots and the "bgr diag" (the Gelman-Rubin R) for all of `mu`, `sigma`, and `y` from the very start of the simulation through, say 1000 iterations. Do 1000 iterations appear adequate as a burn-in for this simulation? Do 10,000 iterations appear adequate as a burn-in?. After enough updates that you are satisfied that burn-in has occurred, run 100,000 more iterations and based on those 100,000 iterations look at "density" and "stats" for all of all of `mu`, `sigma`, and `y`. Based on the 2.5% and 97.5% points of the posterior distributions, what are approximately 95% posterior "credible intervals" for the mean, standard deviation, and an additional part dimension? Are these comparable to your answers from (a)?

(c) GCS&R on page 74 suggest what they call a "non-informative prior" for (μ,σ^2). There is a second set of code below that can be used to at least approximately implement this suggestion in `WinBUGS`. (The extra .00001 in the expression for τ is there because `WinBUGS` won't allow the normal precision to get too small (and for some reason, although it doesn't balk at the model statement for (b), there seem to be numerical problems without adding this insurance against τ too small here), and the $U(-100,100)$ distribution is used instead of the improper "uniform $(-\infty,\infty)$" prior for $\ln\sigma$.) Repeat part (b) using this code. Are your answers here comparable to those in part (b)?

(d) A "diffuse" proper prior for (μ,σ^2) could (for example) be made by using a normal prior with mean 0 and precision .0001 for μ and a gamma prior with mean 1 and variance 100 for τ. How do inferences under such a prior compare to those for parts (a), (b), and (c)?

```
model {
for (i in 1:5) {
x[i]~dnorm(mu, tau)
 }
y~dnorm(mu,tau)
mu~dflat()
tau~dunif(0,10000)
sigma<-sqrt(1/tau)
}
list(x=c(4,3,3,2,3),y=NA)
list(mu=0,tau=1,y=0)
list(mu=500,tau=.0001,y=500)
list(mu=500,tau=10000,y=500)
```

```
model {
U~dunif(-100,100)
tau<-exp(-2.0*U)+.00001
for (i in 1:5) {
x[i]~dnorm(mu, tau)
 }
y~dnorm(mu,tau)
mu~dflat()
sigma<-sqrt(1/tau)
}
list(x=c(4,3,3,2,3),y=NA)
list(mu=0,U=0,y=0)
list(mu=500,U=-100,y=500)
list(mu=500,U=100,y=500)
```

6. What is potentially a fairly serious shortcoming of the analysis in problem 5 above is that the data are really very "coarse"/"granular" relative to the range of the data. That is, although strictly speaking our model says that we have realizations from a (continuous!) normal distribution, we have only 3 distinct values in the (integer) data set, and these have with range of only 2. A possible improvement in the analyses of problem 5 is to think that while there are perhaps unobservable actual part measurements that are "real numbers" (with infinite numbers of digits in their decimal expression) what we actually observe are only "integer-rounded"/"interval-censored" versions of these. (This kind of thinking is perhaps especially sensible in the day of digital instruments that explicitly read to some number of decimal places.) There probably are a couple of ways of building this kind of modeling into a WinBUGS analysis.

   In "pencil and paper" terms, we would want our likelihood (for the integer x_i) to be

$$f(x_1, x_2, x_3, x_4, x_5|\mu, \sigma) = \prod_{i=1}^{5} \left(\Phi\left(\frac{x_i + .5 - \mu}{\sigma} \right) - \Phi\left(\frac{x_i - .5 - \mu}{\sigma} \right) \right)$$

   (for Φ the standard normal cdf) and I'm pretty sure that it's possible to implement this directly using the "zeros trick" (see "Tricks: Advanced Use of the BUGS Language" in the WinBUGS user manual).

   Another method is to use the WinBUGS approach to representing "censoring" (see the "Model Specification" section of the WinBUGS user manual and Section 3.8 of the 2008 course outline). Some code for this parallel to that in 5b) is below. This code treats the real measurements as normal, but unobserved, except to the degree that we know their integer-rounded values. Use this code and compare your inferences about the normal parameters and an additional predicted/unobserved observation to those in 5b). (Note, by the way, that your conclusions here should be consistent with the bit of "statistical folklore" that says that ignoring this rounding issue will, at least for normal observations, tend to artificially inflate one's perception of σ. Note too that handling this analysis in a non-Bayesian/frequentist fashion, i.e. coming up with confidence and prediction intervals, is a non-trivial matter.)

```
model {
x1~dnorm(mu,tau)I(3.5,4.5)
x2~dnorm(mu,tau)I(2.5,3.5)
x3~dnorm(mu,tau)I(2.5,3.5)
x4~dnorm(mu,tau)I(1.5,2.5)
x5~dnorm(mu,tau)I(2.5,3.5)
y~dnorm(mu,tau)
w<-round(y)
mu~dflat()
tau~dunif(0,10000)
sigma<-sqrt(1/tau)
}
list(x1=NA,x2=NA,x3=NA,x4=NA,x5=NA,y=NA)
list(mu=0,tau=1,x1=4,x2=3,x3=3,x4=2,x5=3,y=0)
list(mu=500,tau=.0001,x1=4,x2=3,x3=3,x4=2,x5=3,y=500)
list(mu=500,tau=10000,x1=4,x2=3,x3=3,x4=2,x5=3,y=500)
```

7. Consider again the context of Problem #1. Below are a series of 10 Poisson observations, the first 3 of which were generated using a mean of $\lambda = 2$, the last 7 of which were generated using a mean of $\lambda = 6$.

$$1, 4, 2, 4, 7, 4, 5, 7, 6, 4$$

Suppose that neither the number of observations from the two different means, nor the values of those means were known and one wished to do a Bayes analysis like that suggested in the problem in the previous assignment to try and infer something about those parameters. This is a "change-point problem" and the discussion of "where the size of a set is a random quantity " in the "Model Specification" section of the `WinBUGS` user manual is relevant to implementing a Bayes analysis of this problem.

(a) Below is some `WinBUGS` code for implementing the analysis hinted at in part (a) of Problem #1, together with 6 different starting vectors for the MCMC. Run this code (using 6 chains) through a million iterations, thin to every 100th or 1000th iteration (to speed up the calculations on the output and make the plots easy to see) and interpret the results you obtain.

(b) Change the data set under discussion to

$$1, 4, 2, 4, 7, 4, 50, 70, 60, 40$$

and rerun the code for this problem through 1 million iterations (again using 6 different starting vectors). What indications are there that is a problematic situation for Gibbs sampling? Look in particular at the supposed posterior for the change-point. Does it seem sensible? Rerun the code using only the first 4 starting vectors. Do the `WinBUGS` "answers" seem to change? Make a discussion of this simulation in terms of "islands of probability" and the necessity of seeing many trips between islands in order to sensibly estimate their relative importance. If you think that the analysis in part (a) is more to be trusted than this one, how would you describe the difference between the two cases in qualitative terms?

(c) There is below some `R` code for a Metropolis-Hastings algorithm for the situation in part (b). As far as I can tell, it is OK. Run it with several different starting vectors and make histograms of the simulated values for each of the 3 parameters. All of the runs I've made with the code seem to end up giving the same view of the posterior, namely that with posterior probability very near 1, the change-point is at 6 (this corresponds to one of the 2 islands of probability seen in (b)). It would seem that the second island found in (b) is a very small one! Return to the calculus-type computations referred to in Problem 1a, and with pencil and paper compute the ratio of the posterior probability that $M = 10$ to the posterior probability that $M = 6$.

```
#Here is the WinBUGS code

model {
for (j in 1:2) {
mu[j]~dexp(1)
}
K~dcat(p[])
for (i in 1:10) {
 ind[i]<- 1+step(i-K-.01) #will be 1 for all i<=K, 2 otherwise
 y[i]~dpois(mu[ind[i]])
 }
}
list(p=c(.1,.1,.1,.1,.1,.1,.1,.1,.1,.1),y=c(1,4,2,4,7,4,5,7,6,4))
list(K=1,mu=c(1,10))
list(K=6,mu=c(1,10))
list(K=10,mu=c(1,10))
list(K=1,mu=c(10,1))
list(K=6,mu=c(10,1))
list(K=10,mu=c(10,1))


#Here is the R code


y<-c(1,4,2,4,7,4,50,70,60,40)
sumy<-cumsum(y)
out<-matrix(0,nrow=500000,ncol=3)
cand<-matrix(0,nrow=1,ncol=3)
save<-matrix(0,nrow=500000,ncol=11)
lh<- function(M,mu1,mu2) {
 sumy[M]*log(mu1)+
 (sumy[10]-sumy[M])*log(mu2)-
 ((M+1)*mu1)-((10-M+1)*mu2)
 }
lJ<- function(M,mu1,mu2) {
 -(mu1/20)-(mu2/40)
 }
```

```
out[1,]<-c(10,25,1)
for (i in 2:500000) {
 out[i,]<-out[i-1,]
 cand[1]<-sample(1:10,1)
 cand[2]<-rexp(1,.05)
 cand[3]<-rexp(1,.025)
 accept<-lh(cand[1],cand[2],cand[3])-
 lJ(cand[1],cand[2],cand[3])-
 lh(out[i,1],out[i,2],out[i,3])+
 lJ(out[i,1],out[i,2],out[i,3])
 u<-runif(1,0,1)
 if (log(u)<accept) {
 out[i,1]<-cand[1]
 out[i,2]<-cand[2]
 out[i,3]<-cand[3]}
 save[i,1]<-out[i-1,1]
 save[i,2]<-out[i-1,2]
 save[i,3]<-out[i-1,3]
 save[i,4]<-cand[1]
 save[i,5]<-cand[2]
 save[i,6]<-cand[3]
 save[i,7]<-exp(accept)
 save[i,8]<-u
 save[i,9]<-out[i,1]
 save[i,10]<-out[i,2]
 save[i,11]<-out[i,3]
 }
save[1:20,]
```

Homework #3 (Due March 7, 2008 5PM)

1. Below are $n = 10$ values generated from a normal distribution.

   $$4.90791, 4.83425, 5.19801, 6.85308, 4.07085, 4.66076, 4.16201, 5.49752, 4.15438, 3.72703$$

   Here we consider the problem of Bayes inference for μ and σ based on these data.

   (a) There is some R code below that can be used to make a contour plot for the loglikelihood here, $\ln L(\mu, \sigma)$. Run it and obtain a contour plot for the loglikelihood.

   (b) A number of (joint) prior densities for μ and σ are specified below. Since $g(\mu, \sigma|\boldsymbol{x}) \propto L(\mu, \sigma)\, g(\mu, \sigma)$, it follows that what is essentially a contour plot for log posterior density $\ln g(\mu, \sigma|\boldsymbol{x})$ can be made by contour plotting $\ln g(\mu, \sigma) + \ln L(\mu, \sigma)$. For each of the priors below, make contour plots for both $\ln g(\mu, \sigma)$ AND $\ln g(\mu, \sigma) + \ln L(\mu, \sigma)$. (You can modify the R code below. You will no doubt have to change the "levels" in order to get informative plots. You may type `?contour()`in R to learn more about the contour function.) How would one try to "see" from the contour plots for the log priors that priors i through iii are ones of "independence" and the last two are not?

1. the product of Jeffreys improper priors, $g(\mu, \sigma) = \frac{1}{\sigma^2}$
2. *a priori* $\mu \sim$N$(0, 10^2)$ independent of $\sigma^2 \sim$Inv-$\chi^2(1, 10^2)$
3. *a priori* $\mu \sim$N$(0, 2^2)$ independent of $\sigma^2 \sim$Inv-$\chi^2(3, 2^2)$
4. the jointly conjugate prior discussed in Section 3.3 of the text, for $\mu_0 = 0, \kappa_0 = 1, \nu_0 = 1, \sigma_0 = 2$
5. the jointly conjugate prior discussed in Section 3.3 of the text, for $\mu_0 = 0, \kappa_0 = 10, \nu_0 = 1, \sigma_0 = 2$

```
> gsize<-101
> x<-c(4.90791,4.83425,5.19801,6.85308,4.07085,4.66076,
                     4.16201,5.49752,4.15438,3.72703)
> logL<- function (theta)
+ {
+ sum(log(dnorm(x,theta[1],theta[2])))
+ }
> mu<-3+seq(0,4,length=gsize)
> sigma<-.1+seq(0,2,length=gsize)
> theta<-expand.grid(mu,sigma)
> z<-apply(theta,1,logL)
> z<-matrix(z,gsize,gsize)
> plot(mu,sigma,type="n",main="LogLikelihood")
> contour(mu,sigma,z,levels=c(seq(-30,-10,1)))
```

2. Consider a Bayes analysis of some bivariate normal data, under the complication that not all of $n = 10$ data vectors are complete. In fact, suppose that observations are as in the following table:

x_1	x_2
3.683	-
-	7.324
7.189	6.093
5.014	4.399
4.946	3.540
3.495	-
4.065	2.582
-	5.281
4.222	3.308
-	3.486

There is some WinBUGS code below that can be used to do an analysis assuming that *a priori* the mean vector $\boldsymbol{\mu}$ has

$$\text{mean} \begin{pmatrix} 0 \\ 0 \end{pmatrix} \text{ and covariance matrix } \begin{pmatrix} 100 & 0 \\ 0 & 100 \end{pmatrix}$$

independent of $\mathbf{\Sigma}$ that has

$$\text{mean} \quad \begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$$

and a distribution based on a fairly small number of degrees of freedom.

(a) Argue carefully that the prior used in the `WinBUGS` code has the properties advertised above. Then run the code and find 95% credible intervals for all 5 parameters of the bivariate normal distribution (two means, two standard deviations and a correlation) based on this prior and these data.

(b) Alter the code below as needed to make $\nu = 20$ but hold the same prior mean for $\mathbf{\Sigma}$. Run the code and describe how the inferences change from (a).

(c) Alter the code below as needed to make the prior covariance matrix for $\boldsymbol{\mu}$ the 2×2 identity. Run the code and describe how the inferences change from (a).

(d) Alter the code below as needed to make the prior mean of $\mathbf{\Sigma}$

$$\begin{pmatrix} 4 & -3 \\ -3 & 4 \end{pmatrix}$$

Run the code and describe how the inferences change from (a).

(e) Alter the code below as needed to make the prior mean of $\mathbf{\Sigma}$

$$\begin{pmatrix} 40 & 0 \\ 0 & 40 \end{pmatrix}$$

Run the code and describe how the inferences change from (a)

```
model
   {
        for(i in 1:10)
        {
            Y[i, 1:2] ~dmnorm(mu[], R[ , ])
        }
        mu[1:2] ~dmnorm(alpha[],Tau[ , ])
        R[1:2 , 1:2] ~dwish(Lambda[ , ], nu)
        D[1:2, 1:2]<-inverse(R[1:2, 1:2])
 sig1<-sqrt(D[1,1])
 sig2<-sqrt(D[2,2])
 rho<-D[1,2]/(sig1*sig2)
   }

list(nu=4, alpha=c(0,0),
 Tau = structure(.Data = c(0.01, 0, 0, 0.01), .Dim = c(2, 2)),
 Lambda = structure(.Data = c(4, 0, 0, 4), .Dim = c(2, 2)),
 Y = structure(.Data = c(
                        3.683,NA, NA, 7.324,
                        7.189, 6.093, 5.014, 4.399,
```

```
4.946, 3.540, 3.495, NA,
4.065, 2.582, NA, 5.281,
4.222, 3.308, NA, 3.486), .Dim = c(10, 2)))
```