We have the model
\[ y_{ij} = \mu_i + \epsilon_{ij}, \quad i = 1, \ldots, k; j = 1, \ldots, n_i \]
where \( k = 3, n_1 = 9, n_2 = 11, n_3 = 14 \).

We thus have nine parameters of interest, i.e., \( \theta = (\mu_1, \mu_2, \mu_3, \sigma_1, \sigma_2, \sigma_3, \psi, \tau^2, \sigma) \). The full conditional distributions are given.

We obtain the initial estimates as follows:
\[
\begin{align*}
\hat{\mu}_i &= \bar{y}_i \\
\hat{\sigma}_i^2 &= s_i^2 \\
\hat{\psi} &= \bar{y} \\
\hat{\tau}^2 &= \text{variance of } \bar{y}_1, \bar{y}_2, \bar{y}_3 \\
\hat{\sigma}^2 &= E(\dot{\sigma}^2) = \frac{f_0g_0}{4} = 0.025
\end{align*}
\]

We run 20,000 simulations (Gibbs sampling with burn-in of 500). The MATLAB code is given in Appendix-1. The arithmetic means and medians of the 9 components of \( \theta \) are given in Table 1. The last 500 runs and histograms for each parameter are shown in figures 1-6.

Comparing the means and medians of the posterior distributions of \( \mu_1, \mu_2 \) and \( \mu_3 \), we can say that possibly there exist some difference among the three nematocides.
### Table 1: Means and medians of parameters obtained after 20,000 simulations with burn-in of 500

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>Median</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu_1$</td>
<td>24.0135</td>
<td>23.9336</td>
<td>See Fig-1 and Fig-2 (upper panels)</td>
</tr>
<tr>
<td>$\mu_2$</td>
<td>20.6718</td>
<td>20.6438</td>
<td>See Fig-1 and Fig-2 (middle panels)</td>
</tr>
<tr>
<td>$\mu_3$</td>
<td>21.7825</td>
<td>21.7787</td>
<td>See Fig-1 and Fig-2 (lower panels)</td>
</tr>
<tr>
<td>$\psi$</td>
<td>21.8027</td>
<td>21.6854</td>
<td>See Fig-3 and Fig-4 (upper panels)</td>
</tr>
<tr>
<td>$\tau^2$</td>
<td>16.7927</td>
<td>8.2873</td>
<td>Some large outliers; histogram truncated below 30</td>
</tr>
<tr>
<td>$\sigma^2$</td>
<td>29.9906</td>
<td>25.0127</td>
<td>See Fig-3 and Fig-4 (lower panels)</td>
</tr>
<tr>
<td>$\sigma^2_1$</td>
<td>158.5871</td>
<td>133.5621</td>
<td>See Fig-5 and Fig-6 (upper panels)</td>
</tr>
<tr>
<td>$\sigma^2_2$</td>
<td>111.1912</td>
<td>97.6958</td>
<td>See Fig-5 and Fig-6 (middle panels)</td>
</tr>
<tr>
<td>$\sigma^2_3$</td>
<td>84.2013</td>
<td>76.0645</td>
<td>See Fig-5 and Fig-6 (lower panels)</td>
</tr>
</tbody>
</table>

Figure 1: Final 500 simulations of $\mu_1, \mu_2, \mu_3$

Figure 2: Histograms of $\mu_1, \mu_2, \mu_3$
Figure 3: Final 500 simulations of $\psi, \tau^2, \sigma^2$

Figure 4: Histograms of $\psi, \tau^2, \sigma^2$

Figure 5: Final 500 simulations of $\sigma_1^2, \sigma_2^2, \sigma_3^2$

Figure 6: Histograms of $\sigma_1^2, \sigma_2^2, \sigma_3^2$
2 Problem 2

2.1 Part (a)

We have,

\[ \pi(\theta | y) = \pi(\mu, \sigma | y) = f(y | \mu, \sigma) \pi(\mu) \pi(\sigma) \] by independence of \( \mu \) and \( \sigma \)

\[ \propto \frac{1}{\sigma^n} \exp \left( - \sum_{i=1}^{n} \left\{ \frac{y_i - \mu}{\sigma} + \exp \left( - \frac{y_i - \mu}{\sigma} \right) \right\} \right) \exp \left( - \frac{\mu^2}{200} \right) \frac{1}{\sigma} \exp \left( - \frac{(\ln \sigma)^2}{200} \right) \]

2.2 Part (b)

In order to construct the proposal distribution for the Metropolis-Hastings algorithm, we consider \( \mu \sim N(\mu_n, s_1^2) \) and \( \sigma \sim LN(\ln \sigma_n, s_2^2) \), and utilizing their independence, we obtain the proposal distribution as

\[ q(\theta | \theta_n) = q(\mu, \sigma | \mu_n, \sigma_n) \propto \frac{1}{s_1 s_2 \sigma} \exp \left[ - \frac{1}{2} \left\{ \frac{(\mu - \mu_n)^2}{s_1^2} + \frac{(\ln \sigma - \ln \sigma_n)^2}{s_2^2} \right\} \right] \]

Note that the suffix \( n \) denotes the iteration number during simulation.

Suppose \( \theta_n \) denotes the current value of \( \theta \) after \( n \) iterations and at the \( (n+1)^{th} \) stage, we generate \( \theta_{n+1} \) using the proposal distribution stated above. Then,

\[ \frac{q(\theta_n | \theta_{n+1})}{q(\theta_{n+1} | \theta_n)} = \frac{\sigma_{n+1}}{\sigma_n} \] and

\[ \rho(\theta_n, \theta_{n+1}) = \frac{q(\theta_n | \theta_{n+1}) \pi(\mu_{n+1}, \sigma_{n+1})}{q(\theta_{n+1} | \theta_n) \pi(\mu_n, \sigma_n)} = \frac{\sigma_{n+1}}{\sigma_n} \frac{\pi(\mu_{n+1}, \sigma_{n+1})}{\pi(\mu_n, \sigma_n)} \]

where \( \pi(\mu, \sigma) \) is as defined in part(a).

The Metropolis Hastings algorithm is thus developed as follows:

1. Draw a random number \( U \) from \( U(0, 1) \).
2. Draw \( \theta_{n+1} \) from the proposal distribution.
3. Accept \( \theta_{n+1} \) as the new value of \( \theta \) if \( U < \rho \land 1 \), or equivalently if \( \log(U) < \log(\rho) \land 0 \), where \( \rho \) is as defined above.
The MATLAB code is given in Appendix-2. 40,000 simulations with a burn-in of 2000 were run. Figures 7 and 8 show respectively the histograms and last 500 simulations. It is obvious from figure 8 that the algorithm was not very smooth; however reasonable distributions of $\mu$ and $\sigma$ were obtained, as seen in Figure 7. The summary statistics are given in Table-2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>Median</th>
<th>s.d.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>45.6157</td>
<td>45.7444</td>
<td>3.2113</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>21.1014</td>
<td>20.9445</td>
<td>2.6133</td>
</tr>
</tbody>
</table>

2.3 Part(c)

The Bayes’ estimators of $\mu$ and $\sigma$ are obtained from Table 2 (the means of their posterior distributions) as $\hat{\mu}_{Bayes} = 45.6157$ and $\hat{\sigma}_{Bayes} = 21.1014$.

Thus,

$$P(y^* \geq 410|y) = 1 - F(410|\hat{\mu}_{Bayes}, \hat{\sigma}_{Bayes}) = 1 - \exp \left\{ - \exp \left( -\frac{410 - \hat{\mu}_{Bayes}}{\hat{\sigma}_{Bayes}} \right) \right\} = 3.1659 \times 10^{-8},$$

which is negligibly small. What happened in 1999 was an event of almost zero probability!
# 3 Problem 3

The summary statistics for $\lambda, p_2, p_7, p_9, p_{10}, p_{13}, p_{17}$ are given in the table below. Figures 9-15 show the corresponding histograms obtained by importing the data (output of BUGS) to MATLAB.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Mean</th>
<th>sd</th>
<th>MC error</th>
<th>val 2.5%</th>
<th>median</th>
<th>val 97.5%</th>
<th>start</th>
<th>sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda$</td>
<td>8.635</td>
<td>0.6661</td>
<td>0.005248</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
<td>1001</td>
<td>20000</td>
</tr>
<tr>
<td>$p_2$</td>
<td>0.02038</td>
<td>0.02002</td>
<td>7.655E-5</td>
<td>5.077E-4</td>
<td>0.01424</td>
<td>0.07379</td>
<td>1001</td>
<td>20000</td>
</tr>
<tr>
<td>$p_7$</td>
<td>0.06163</td>
<td>0.03422</td>
<td>1.356E-4</td>
<td>0.01289</td>
<td>0.05562</td>
<td>0.1439</td>
<td>1001</td>
<td>20000</td>
</tr>
<tr>
<td>$p_9$</td>
<td>0.09141</td>
<td>0.04176</td>
<td>1.845E-4</td>
<td>0.02698</td>
<td>0.08601</td>
<td>0.1878</td>
<td>1001</td>
<td>20000</td>
</tr>
<tr>
<td>$p_{10}$</td>
<td>0.1038</td>
<td>0.04347</td>
<td>1.806E-4</td>
<td>0.03548</td>
<td>0.09833</td>
<td>0.2031</td>
<td>1001</td>
<td>20000</td>
</tr>
<tr>
<td>$p_{13}$</td>
<td>0.08157</td>
<td>0.03878</td>
<td>1.626E-4</td>
<td>0.02306</td>
<td>0.07593</td>
<td>0.1723</td>
<td>1001</td>
<td>20000</td>
</tr>
<tr>
<td>$p_{17}$</td>
<td>0.02018</td>
<td>0.01983</td>
<td>8.023E-5</td>
<td>5.363E-4</td>
<td>0.01409</td>
<td>0.07381</td>
<td>1001</td>
<td>20000</td>
</tr>
</tbody>
</table>

![Figure 9: Histogram of $\lambda$](image)
Figure 10: Histogram of $p_2$  

Figure 11: Histogram of $p_7$  

Figure 12: Histogram of $p_9$  

Figure 13: Histogram of $p_{10}$
4 APPENDIX 1: MATLAB CODE FOR PROBLEM

function unbalanced_anova
% mcmc2.m
%%%%%%%%%%%%%%%%
% Y_ij = mu_i + epsilon_ij, i=1,...,k;
% j=1,...,n_i.
% mu_i ~ N(psi, tau^2)
% epsilon_ij ~ N(0, sigma_i^2)
% psi ~ N(psi_0, tau^2/zeta_0)
% sigma_i^2 ~ IG(a_0/2, a0*sigma^2/2)
% tau^2 ~ IG(c_0/2, d_0/2)
% sigma^2 ~ gamma(f_0/2, g_0/2)
%-----
clear all close all rand('state',2); randn('state',2);
%-----------------figure defaults
lw = 2; set(0, 'DefaultAxesFontSize', 17); fs = 14; msize = 5;
%----------------Here we enter the data-------------
k=3; n1=9; n2=11; n3=14;
y1=[25.8,19.8,28.6,29.4,22.3,33.8,33.8,27.8,29.6];
y2=[16.5,23.5,13.5,34.6,16.9,18.8,26.1,18.4,17.2,11.6,20.2];
y3=[24.0,29.1,16.0,24.8,27.0,10.9,11.8,23.2,17.7,23.9,24.6,24.0,27.2,23.7];
y=[y1,y2,y3];

%------------Initial computations ---------------------
n=n1+n2+n3; ybar=mean(y1); y2bar=mean(y2); y3bar=mean(y3);
ybars = [ybar, y2bar, y3bar]; % vector of ybars
s1=std(y1); s2=std(y2); s3=std(y3); ybar=mean(y); s=std(y);
sig=std(ybars);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
M = 20000; bin=100; mu1s = []; mu2s = []; mu3s = []; sigma1_sqs = [];
sigma2_sqs = []; sigma3_sqs = []; tau_sqs = [];
psi = ybar; sigma_sq = 0.025;

%----------Initializing parameters------------------
a0=1; c0=1; d0=1; f0=1; g0=0.1; psi0=10; zeta0=0.1; mu1 = y1bar;
mu2 = y2bar; mu3 = y3bar; sigma1_sq = s1^2; sigma2_sq = s2^2;
sigma3_sq = s3^2; psi = ybar; tau_sq = sig^2; sigma_sq = 0.025;

%--------------Repeated simulations-----------------------------
h=waitbar(0,'Simulation in progress');
for r = 1: M

mu1 = 1/sqrt(n1/sigma1_sq + 1/tau_sq)*randn + ...
(n1*y1bar/sigma1_sq + psi/tau_sq)/(n1/sigma1_sq + 1/tau_sq);
mu2 = 1/sqrt(n2/sigma2_sq + 1/tau_sq)*randn + ...
(n2*y2bar/sigma2_sq + psi/tau_sq)/(n2/sigma2_sq + 1/tau_sq);
mu3 = 1/sqrt(n3/sigma3_sq + 1/tau_sq)*randn + ...
(n3*y3bar/sigma3_sq + psi/tau_sq)/(n3/sigma3_sq + 1/tau_sq);

psi = sqrt(tau_sq/(k+zeta0))*randn + (mu1+mu2+mu3+zeta0*psi0)/(k+zeta0);

tau_sq = (gamrnd((c0+k+1)/2, 2/(d0 + (mu1-psi)^2 + (mu2-psi)^2 + ...
(mu3-psi)^2)*(k+zeta0))^{-1};

sigma1_sq = (gamrnd((a0+n1)/2,2/(a0*sigma_sq + ...
(n1-1)*s1^2 + (n2-1)*s2^2 + (n3-1)*s3^2 + n1*(y1bar-mu1)^2)))^{-1};

sigma2_sq = (gamrnd((a0+n2)/2,2/(a0*sigma_sq + ...
(n1-1)*s1^2 + (n2-1)*s2^2 + (n3-1)*s3^2 + n2*(y2bar-mu2)^2)))^{-1};

sigma3_sq = (gamrnd((a0+n3)/2,2/(a0*sigma_sq + ...
(n1-1)*s1^2 + (n2-1)*s2^2 + (n3-1)*s3^2 + n3*(y3bar-mu3)^2)))^{-1};

sigma_sq = gamrnd((f0+k*a0)/2,2/(g0+a0*(1/sigma1_sq+1/sigma2_sq+1/sigma3_sq)));

%Storage
mu1s = [mu1s mu1];
mu2s = [mu2s mu2];
mu3s = [mu3s mu3];
psi = [psi psi];

waitbar(r/M)
end close(h)
%----------------Plotting runchart--------------------------
figure(1) subplot(3,1,1) plot((M-500:M), mu1s(M-500:M)) subplot(3,1,2) plot((M-500:M), mu2s(M-500:M)) subplot(3,1,3) plot((M-500:M), mu2s(M-500:M))

figure(2) subplot(3,1,1) plot((M-500:M), psis(M-500:M)) subplot(3,1,2) plot((M-500:M), tau_sqs(M-500:M)) subplot(3,1,3) plot((M-500:M), sigma_sqs(M-500:M))

figure(3) subplot(3,1,1) plot((M-500:M), sigma1_sqs(M-500:M)) subplot(3,1,2) plot((M-500:M), sigma2_sqs(M-500:M)) subplot(3,1,3) plot((M-500:M), sigma3_sqs(M-500:M))

%-----------Plotting Histograms--------------------------------
figure(4) burnin = 2000; subplot(3,1,1) hist(mu1s(burnin:M),bin) subplot(3,1,2) hist(mu2s(burnin:M),bin) subplot(3,1,3) hist(mu3s(burnin:M),bin)

figure(5) subplot(3,1,1) hist(psis(burnin:M),bin) subplot(3,1,2) tau_sqss=tau_sqs(burnin:M); tau_sqsss=tau_sqss(tau_sqss<30); hist(tau_sqsss,bin) subplot(3,1,3) hist(sigma_sqs(burnin:M),bin)

figure(6) subplot(3,1,1) hist(sigma1_sqs(burnin:M),bin) subplot(3,1,2) hist(sigma2_sqs(burnin:M),bin) subplot(3,1,3) hist(sigma3_sqs(burnin:M),bin)

%-------------Obtaining means and medians-------------------------------
meanmu1=mean(mu1s(burnin:M)) meanmu2=mean(mu2s(burnin:M)) meanmu3=mean(mu3s(burnin:M)) meanpsi=mean(psis(burnin:M))
meantau_sq=mean(tau_sqs(burnin:M))
meansigma1_sq=mean(sigma1_sqs(burnin:M)) meansigma2_sq=mean(sigma2_sqs(burnin:M)) meansigma3_sq=mean(sigma3_sqs(burnin:M)) meansigma_sq=mean(sigma_sqs(burnin:M))

medianmu1=median(mu1s(burnin:M)) medianmu2=median(mu2s(burnin:M)) medianmu3=median(mu3s(burnin:M)) medianpsi=median(psis(burnin:M))
mediantau_sq=median(tau_sqs(burnin:M))
mediansigma1_sq=median(sigma1_sqs(burnin:M)) mediansigma2_sq=median(sigma2_sqs(burnin:M)) mediansigma3_sq=median(sigma3_sqs(burnin:M)) mediansigma_sq=median(sigma_sqs(burnin:M))
function metropolis
% Given mu, sigma, y follows a Gumbel distribution.
% Prior on mu is N(0,10^2) and that on sigma is LN(0,10^2).
clear all close all
%-----------------figure defaults
lw = 2; set(0, 'DefaultAxesFontSize', 15); fs = 14; msize = 5;
randn('seed',3)
%--------------------------------
nn = 40000; % nn=number of metropolis iterations
s1=2; s2=1; % proposals: mu ~ N(mu_n,s1^2), sigma ~ LN(sigma_n, s2^2)
%s=0.1
burn=2000; % burn = burnin amount
%----------------Entering the data-----------------------------
y=[154,49.6,46.7,58.3,70.5,90,70.1,105.7,37.4,40.8,34.7,72.2,30,71.6,100,...
33.7,49.9,56.1,142.3,28.6,54.8,74.1,60,50.9,38.6,53.4,132.5,50.7,40.8,84.3,...
38.8,27.4,67,118.7,23.2,55,67.9,87.3,39,98.7,47.1,71.6,83.6,44.3,41.2,35.9,44.3]
%-------------------Initialization-----------------------------
mus=[]; sigmas=[]; oldmu = 0; oldsigma =1;
%------------------Simulations-------------------------------
h=waitbar(0,'Simulation in progress'); for i = 1:nn
newmu = oldmu + s1*randn; %proposal from N(oldmu, s1^2)
newlogsigma = log(oldsigma) + s2*randn; %proposal from LN(log(oldsigma), s2^2)
newsigma=exp(newlogsigma);
u = rand;
l_numr=log(newsigma)-49*log(newsigma)-sum(((y-newmu)/newsigma)+ ...
(exp((newmu-y)/newsigma)))-(newmu^2+(log(newsigma))^2)/200;

l_denr=log(oldsigma)-49*log(oldsigma)-sum(((y-oldmu)/oldsigma)+ ...
(exp((oldmu-y)/oldsigma)))-(oldmu^2+(log(oldsigma))^2)/200;
lp=l_numr-l_denr;
if log(u) <= min(lp, 0)
oldmu = newmu; oldsigma=newsigma; % set 'old' to be the 'new' for the next iteration;
end
mus = [mus, oldmu]; sigmas = [sigmas, oldsigma]; %collect all new mus and sigmas
waitbar(i/nn)
end
figure(1)
subplot(2,1,1)
hist(mus(1+burn:nn),50)
subplot(2,1,2)
hist(sigmas(1+burn:nn),50)
figure(2)
subplot(2,1,1)
plot(mus(nn-500:nn))
subplot(2,1,2)
plot(sigmas(nn-500:nn));

mean_mu=mean(mus(burn:nn))
median_mu=median(mus(burn:nn))
sd_mu=std(mus(burn:nn))
mean_sigma=mean(sigmas(burn:nn))
median_sigma=median(sigmas(burn:nn))
sd_sigma=std(sigmas(burn:nn))