ISyE 8843, Mid-term Exam

Tirthankar Dasgupta

1 Problem 1

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:

$$\begin{aligned} \hat{\mu_i} &= \bar{y_{i.}} \\ \hat{\sigma_i^2} &= s_i^2 \\ \hat{\psi} &= \bar{y_{..}} \\ \hat{\tau^2} &= \text{variance of } \bar{y_{1.}}, \bar{y_{2.}}, \bar{y_{3.}} \\ \hat{\sigma^2} &= \widehat{E(\sigma^2)} = \frac{f_0 g_0}{4} = 0.025 \end{aligned}$$

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 θ 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 μ_1, μ_2 and μ_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

Parameter	Mean	Median	Remarks
μ_1	24.0135	23.9336	See Fig-1 and Fig-2 (upper panels)
μ_2	20.6718	20.6438	See Fig-1 and Fig-2 (middle panels)
μ_3	21.7825	21.7787	See Fig-1 and Fig-2 (lower panels)
ψ	21.8027	21.6854	See Fig-3 and Fig-4 (upper panels)
τ^2	16.7927	8.2873	Some large outliers; histogram truncated below 30 See Fig-3 and Fig-4 (middle panels)
σ^2	29.9906	25.0127	See Fig-3 and Fig-4 (lower pannels)
σ_1^2	158.5871	133.5621	See Fig-5 and Fig-6(upper pannels)
σ_2^2	111.1912	97.6958	See Fig-5 and Fig-6 (middle pannels)
σ_3^2	84.2013	76.0645	See Fig-5 and Fig-6 (lower pannels)

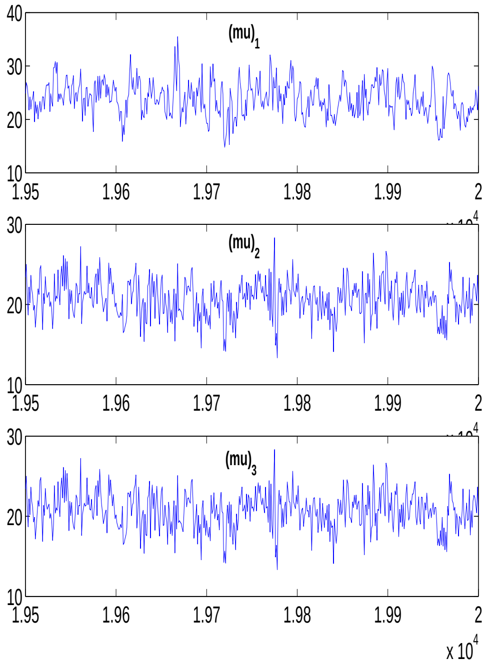


Figure 1 : Final 500 simulations of μ_1,μ_2, μ_3

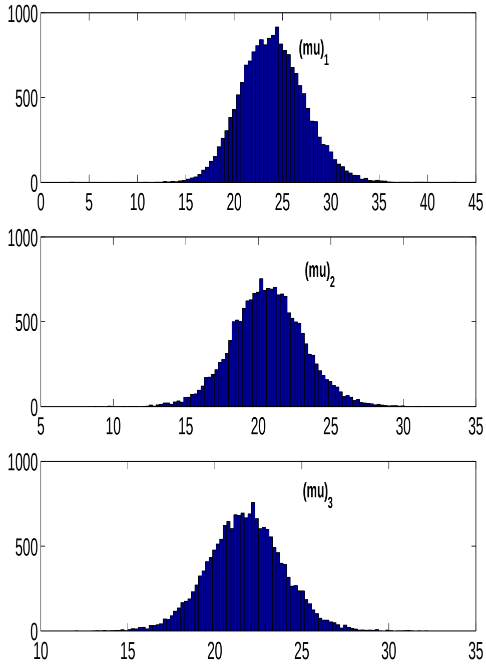
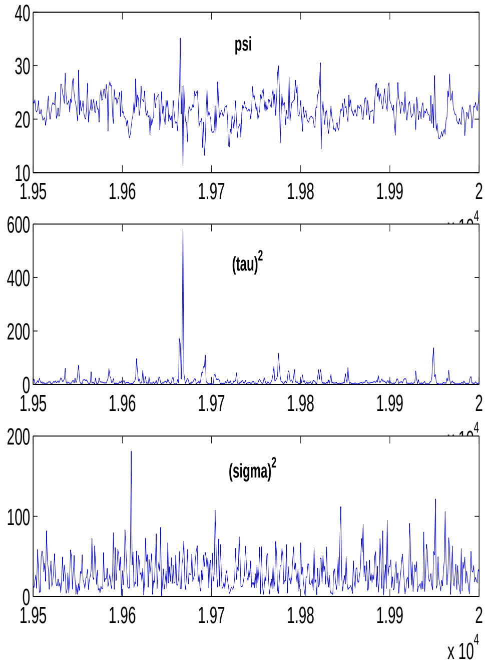


Figure 2 : Histograms of μ_1,μ_2, μ_3



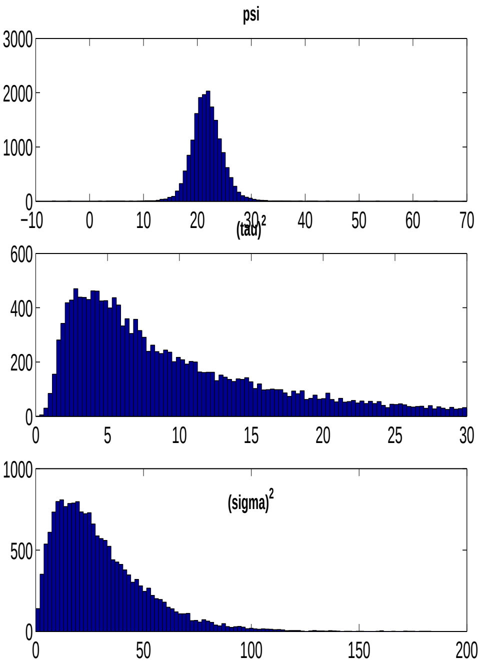
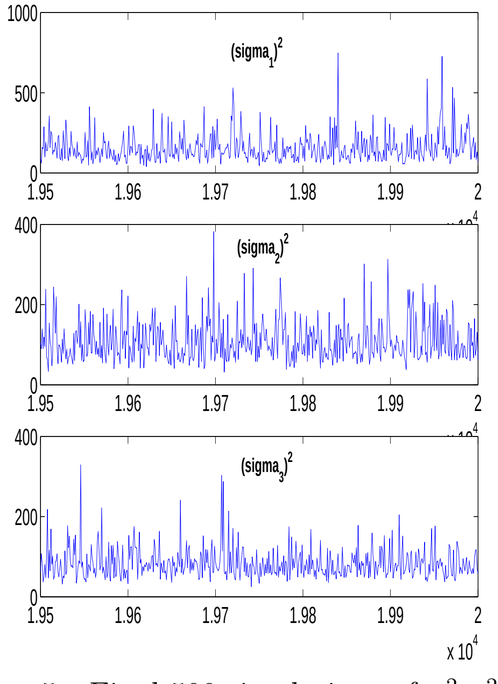


Figure 3 : Final 500 simulations of ψ,τ^2, σ^2 Figure 4 : Histograms of ψ,τ^2, σ^2



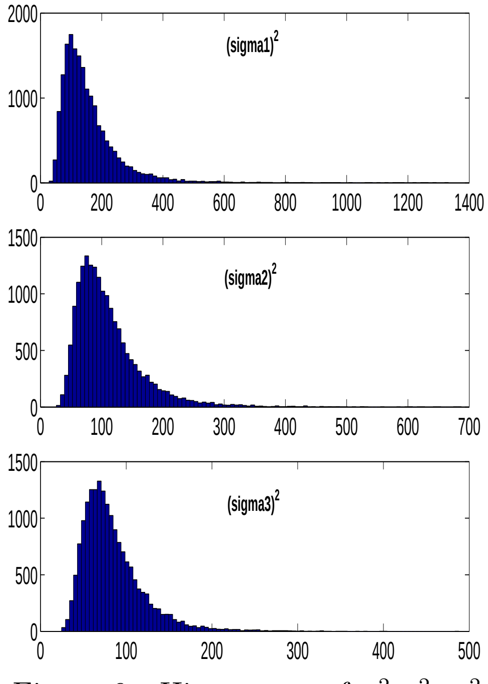


Figure 5 : Final 500 simulations of σ_1^2,σ_2^2, σ_3^2 Figure 6 : Histograms of σ_1^2,σ_2^2, σ_3^2

2 Problem 2

2.1 Part (a)

We have,

$$\begin{aligned}
\pi(\theta|y) &= \pi(\mu,\sigma|y) \\
&= f(\mathbf{y}|\mu,\sigma)\pi(\mu)\pi(\sigma) \;\text{ by independence of } \mu \text{ and } \sigma \\
&\propto \frac{1}{\sigma^n}\exp\Bigg(-\sum_{i=1}^{n}\Big\{\frac{y_i-\mu}{\sigma}+\exp\Big(-\frac{y_i-\mu}{\sigma}\Big)\Big\}\Bigg)\exp\Big(-\frac{\mu^2}{200}\Big)\frac{1}{\sigma}\exp\Big(-\frac{(\ln\sigma)^2}{200}\Big) \\
&\propto \frac{1}{\sigma^{n+1}}\exp\Bigg(-\sum_{i=1}^{n}\Big\{\frac{y_i-\mu}{\sigma}+\exp\Big(-\frac{y_i-\mu}{\sigma}\Big)\Big\}-\frac{\mu^2+(\ln\sigma)^2}{200}\Bigg)
\end{aligned}$$

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

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

Note that the suffix n denotes the iteration number during simulation.

Suppose θ_n denotes the current value of θ after n iterations and at the $(n+1)^{th}$ stage, we generate θ_{n+1} using the proposal distribution stated above. Then,

$$\begin{aligned}
\frac{q(\theta_n|\theta_{n+1})}{q(\theta_{n+1}|\theta_n)} &= \frac{\sigma_{n+1}}{\sigma_n} \;\text{ and} \\
\rho(\theta_n,\theta_{n+1}) &= \frac{q(\theta_n|\theta_{n+1})}{q(\theta_{n+1}|\theta_n)}.\frac{\pi(\mu_{n+1},\sigma_{n+1})}{\pi(\mu_n,\sigma_n)} = \frac{\sigma_{n+1}}{\sigma_n}.\frac{\pi(\mu_{n+1},\sigma_{n+1})}{\pi(\mu_n,\sigma_n)}
\end{aligned}$$

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 θ_{n+1} from the proposal distribution.
3. Accept θ_{n+1} as the new value of θ if $U < \rho \bigwedge 1$, or equivalently if $\log(U) < \log(\rho) \bigwedge 0$, where ρ 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 μ and σ were obtained, as seen in Figure 7. The summary statistics are given in Table-2.

Table 2: Means, medians and standard deviations of μ and σ

Parameter	Mean	Median	s.d.
μ	45.6157	45.7444	3.2113
σ	21.1014	20.9445	2.6133

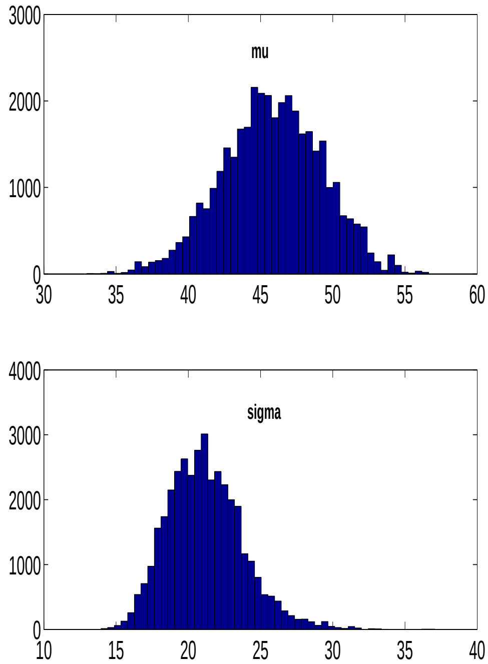


Figure 7 : Histograms of μ and σ

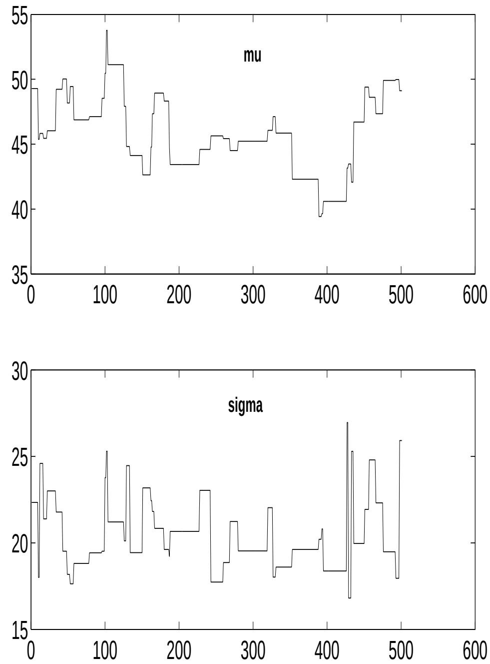


Figure 8 : Final 500 simulations of μ andσ

2.3 Part(c)

The Bayes' estimators of μ and σ 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 Problem3

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 3: Posterior statistics of all parameters

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

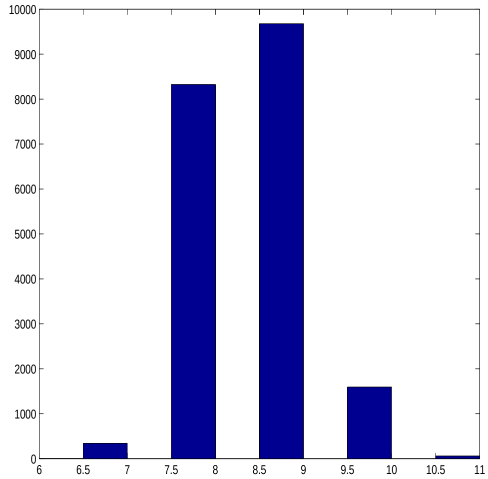


Figure 9 : Histogram of λ

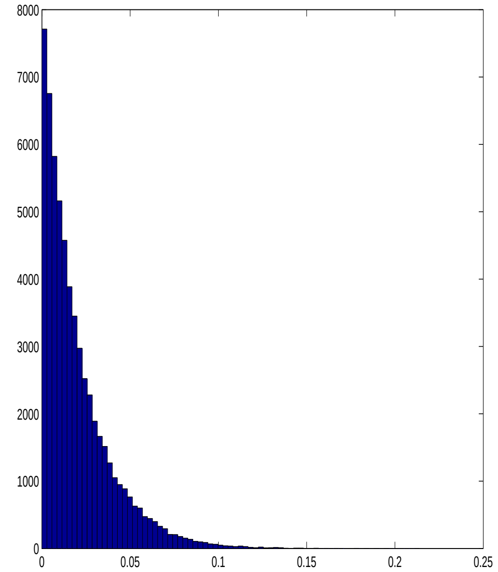


Figure 10 : Histogram of p_2

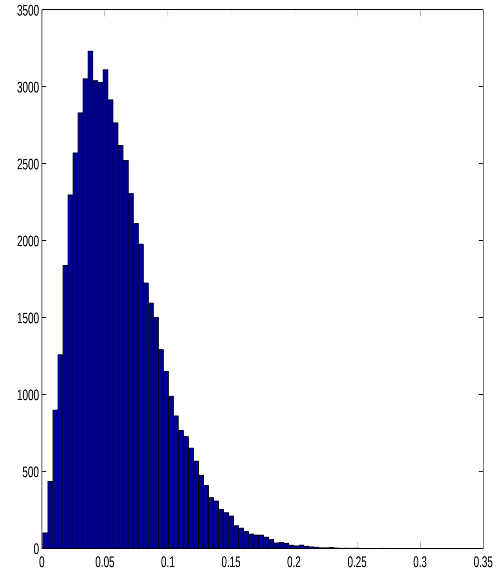


Figure 11 : Histogram of p_7

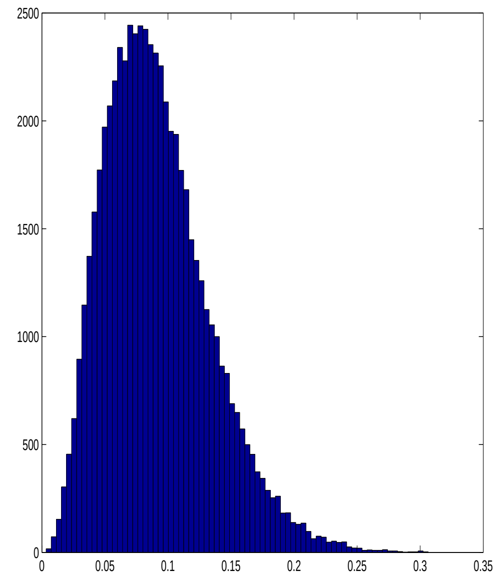


Figure 12 : Histogram of p_9

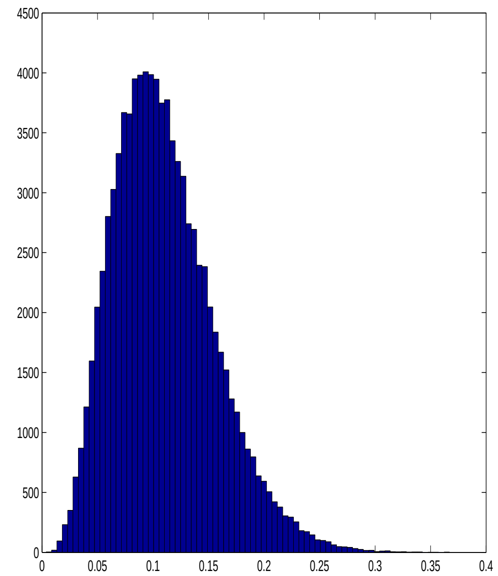


Figure 13 : Histogram of p_{10}

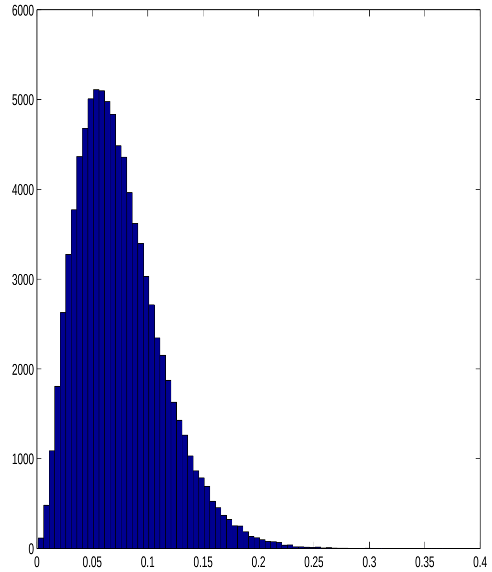


Figure 14 : Histogram of p_{13}

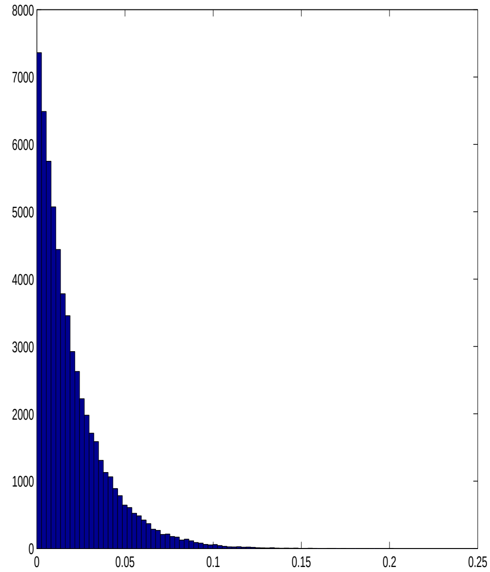


Figure 15 : Histogram of p_{17}

4 APPENDIX 1: MATLAB CODE FOR PROBLEM-1

```
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; y1bar=mean(y1); y2bar=mean(y2); y3bar=mean(y3);
ybars = [y1bar, 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 = []; psis = [];
sigma_sqs =[];

%----------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 + zeta0*(psi-psi0)^2)))^(-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];
 psis = [psis psi];
 tau_sqs = [tau_sqs tau_sq];
 sigma1_sqs = [sigma1_sqs sigma1_sq];
 sigma2_sqs = [sigma2_sqs sigma2_sq];
 sigma3_sqs = [sigma3_sqs sigma3_sq];
 sigma_sqs= [sigma_sqs sigma_sq];

 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))
```

5 APPENDIX 2: MATLAB CODE FOR PROBLEM-2

```
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=[154,49.6,46.7,58.3,70.5,90,70.1,105.7,37.4,40.8,34.7,58.9,72.2,30,71.6,100,...
   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,...
   38.8,27.4,67,118.7,23.2,55,67.9,87.3,89,98.7,47.1,71.6,83.6,44.3,41.2,35.9,44.3]

%-------------------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))
```