**#QUESTION 1**

**#1a)**

Hand-written in a separate pdf, provided in the same email.

**#1b)**  
#Algorithm attached in the appendix

**#1c)**

**#**Plot given below for different temperatures

#R-code

plot(density(mu1[-(1:5)]),type="l", xlab="mu", ylab="Posterior density ",

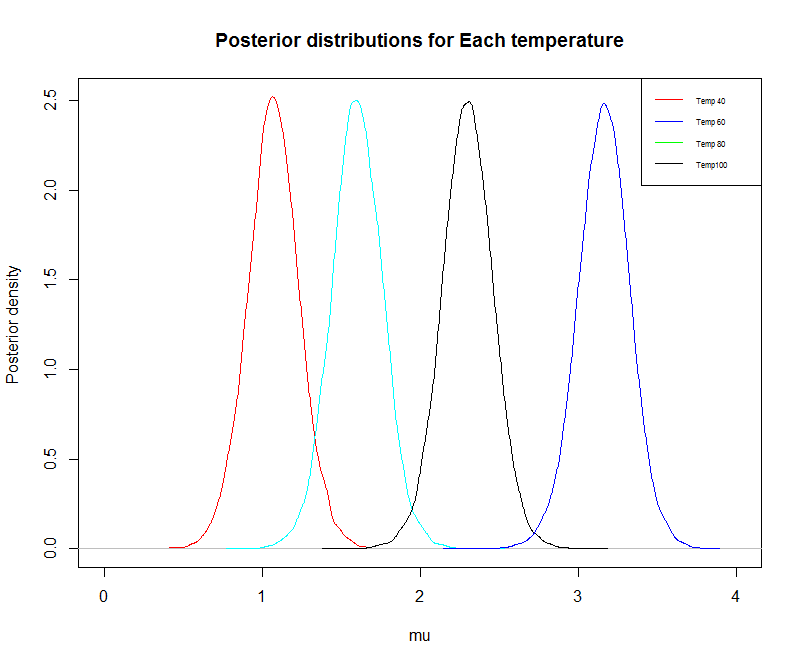
cex=1.5, xlim=c(0,4), col="red", lty=1, main="Posterior distributions for Each temperature")

lines(density(mu2[-(1:5)]), col="cyan",lty=1)

lines(density(mu3[-(1:5)]),col="black",lty=1)

lines(density(mu4[-(1:5)]),col="blue", lty=1)

legend("topright", c("Temp 40", "Temp 60", "Temp 80", "Temp100"),lty=c(1, 1, 1,1), col=c("red","blue","green","black"),cex=0.5)



Some summary statistics for the drug effects means

**#some summary statistics**

#for temperature of 40

> temp1<-rnorm(nbig,mean=mu1,sd=1/sqrt(tau))

> mean(temp1)

[1] 1.069945

> sd(temp1)

[1] 0.4081442

> quantile(temp1, probs=seq(0,1,0.025)) # can use this for quantiles

#for temperature of 60

> temp2<-rnorm(nbig,mean=mu2,sd=1/sqrt(tau))

> mean(temp2)

[1] 1.600204

> sd(temp2)

[1] 0.4062412

> quantile(temp2, probs=seq(0,1,0.025)) # can use this for quantiles

#for temperature of 80

> temp3<-rnorm(nbig,mean=mu3,sd=1/sqrt(tau))

> mean(temp3)

[1] 2.30398

> sd(temp3)

[1] 0.4087044

> quantile(temp3, probs=seq(0,1,0.025)) # can use this for quantiles

#for temperature of 100

> temp4<-rnorm(nbig,mean=mu4,sd=1/sqrt(tau))

> mean(temp4)

[1] 3.166451

> quantile(temp4, probs=seq(0,1,0.025)) # can use this for quantiles

> sd(temp4)

[1] 0.4094899

**#1d)**

#R-code and plot given below

Posterior probability will be 0, since temp3-temp1 is always less than 0 at all occasions, as indicated by the quantiles and the plot below

> newmu<-mu1-mu4

> mean(temp1-temp3)

[1] -1.234035

> sd(temp1-temp3)

[1] 0.5759707

> quantile(temp1-temp3, probs=seq(0,1,0.025))

0% 2.5% 5% 7.5% 10% 12.5% 15% 17.5%

-4.12531929 -2.37636800 -2.17663344 -2.05442578 -1.96330131 -1.88277242 -1.81614243 -1.75787869

20% 22.5% 25% 27.5% 30% 32.5% 35% 37.5%

-1.70532621 -1.65623270 -1.60990823 -1.56224928 -1.52216557 -1.48186975 -1.44481164 -1.40654462

40% 42.5% 45% 47.5% 50% 52.5% 55% 57.5%

-1.37231110 -1.33635167 -1.30237367 -1.26755847 -1.23397548 -1.20190442 -1.16700737 -1.13250013

60% 62.5% 65% 67.5% 70% 72.5% 75% 77.5%

-1.09699369 -1.05949419 -1.02266110 -0.98388196 -0.94512866 -0.90241463 -0.85663694 -0.81221161

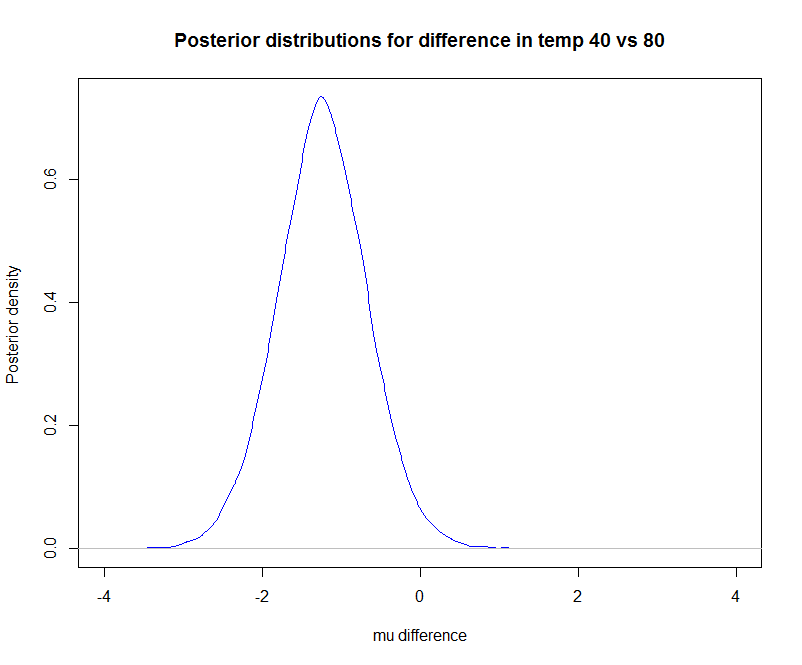
80% 82.5% 85% 87.5% 90% 92.5% 95% 97.5%

-0.76339043 -0.71119758 -0.65730540 -0.59085445 -0.50731054 -0.41805849 -0.28787178 -0.09496114

100%

1.55223010

> plot(density(temp1-temp3),type="l", xlab="mu difference", ylab="Posterior density ", cex=1.5, xlim=c(-4,4), col="blue", lty=1, main="Posterior distributions for difference in temp 40 vs 80")



> mean(temp1>temp3)

[1] 0.0178

**#Question 2)**

**#2a**

#Reading the csv file for import the data

smokeDat=read.table("SmokeAgeDeath.csv",header=TRUE,sep = ",")

#**Code for OpenBUGS model given below**

cat("

model{

for(i in 1:20)

{

#smoke age death pyears

death[i]~dpois(lam[i])

log(lam[i]) <- log(pyears[i]) + beta0 + beta.s[smoke[i]] + beta.c[age[i]] + b[i]

b[i] ~dnorm(0,tau)

b.adj[i] <- b[i] - mean(b[])

}

for(is in 1:4){

beta.s[is]~dnorm(0,tau.s)

beta.s.adj[is] <- beta.s[is] -mean(beta.s[])

}

for(ic in 1:5){

beta.c[ic]~dnorm(0,tau.c)

beta.c.adj[ic] <- beta.c[ic] - mean(beta.c[])

}

# Note: total person-years per categories is less than 115,000

# ln(115000) ~ = 11.65....

# so rate has to be bigger than 1/115,000 and log(rate) > -11.65...

# so log( base rate) should be between about -12 and 12.

# 1/12/12 is about .00694

beta0 ~ dnorm(0, .00694)

beta0.adj <- beta0 + mean(b[]) + mean(beta.s[])+ mean(beta.c[])

# for the <extra poisson variation> ...

# assume bounded by very big number... say 1000 times...

# so log(1000) is about 2.3\*4 which is about 9.2

std ~ dunif(0, 9)

tau <- 1/std/std

# for the relative risk between groups... a very large number would be 100 times,

# so, log(100) is about 2.3\*2 or about 4.6

# also, note that 1/5/5 is 0.04

#

std.s ~dunif(0, 5)

tau.s <- 1/std.s/std.s

std.c ~ dunif(0,5)

tau.c <- 1/std.c/std.c

beta.o~dnorm(0, .04)

}", file="smokemod.txt")

**#defining parameters and data for the bugs function**

params=c("beta.s.adj", "std.s", "beta.c.adj", "std.c", "beta0.adj", "std")

attach(smokeDat)

bug.dat=list("smoke","age","death", "pyears")

init.fun=function(){list(

beta.s=rnorm(4), std.s=runif(1,1,2),

beta.c=rnorm(5), std.c=runif(1,1,2),

std=runif(1,1,2), beta0=rnorm(1),

b=rnorm(20,0,.1))}

**#using openBUGS to run our model, this code also gives the posterior which we need for 2a in OpenBUGS**

smokeBug0=bugs(bug.dat, init.fun, params, model.file="smokemod.txt",

n.chains=5, n.iter=8000, n.burnin=1000, debug=TRUE #for production

# n.chains=5, n.iter=6000, n.burnin=1000, debug=TRUE #for testing

)

**#OpenBUGS output showing the posterior distributions**

**#the univariate posteriors of interest are highlighted**

Summary statistics

**mean sd val2.5pc median val97.5pc sample**

beta.c.adj[1] -1.343 0.1718 -1.693 -1.339 -1.017 45000

beta.c.adj[2] -0.721 0.1467 -1.017 -0.7189 -0.4383 45000

beta.c.adj[3] -0.02228 0.1261 -0.2747 -0.02187 0.2202 45000

beta.c.adj[4] 0.6805 0.1192 0.447 0.6812 0.9125 45000

beta.c.adj[5] 1.406 0.09914 1.213 1.405 1.6 45000

beta.s.adj[1] -0.4608 0.09023 -0.6353 -0.4616 -0.2821 45000

beta.s.adj[2] -0.5325 0.1027 -0.7323 -0.5326 -0.3305 45000

beta.s.adj[3] 0.06716 0.1369 -0.2088 0.0707 0.3272 45000

beta.s.adj[4] 0.9261 0.1159 0.6964 0.9275 1.15 45000

beta0.adj -7.373 0.06995 -7.514 -7.372 -7.239 45000

deviance 102.1 4.287 95.66 101.5 112.3 45000

std 0.08483 0.06756 0.01204 0.06602 0.2558 45000

std.c 1.823 0.9956 0.7367 1.429 4.485 45000

std.s 1.335 0.9032 0.4218 1.035 3.903 45000

Deviance information

**Dbar Dhat DIC pD**

death 102.1 92.78 111.4 9.312

total 102.1 92.78 111.4 9.312

**#plots produced in OpenBUGS**









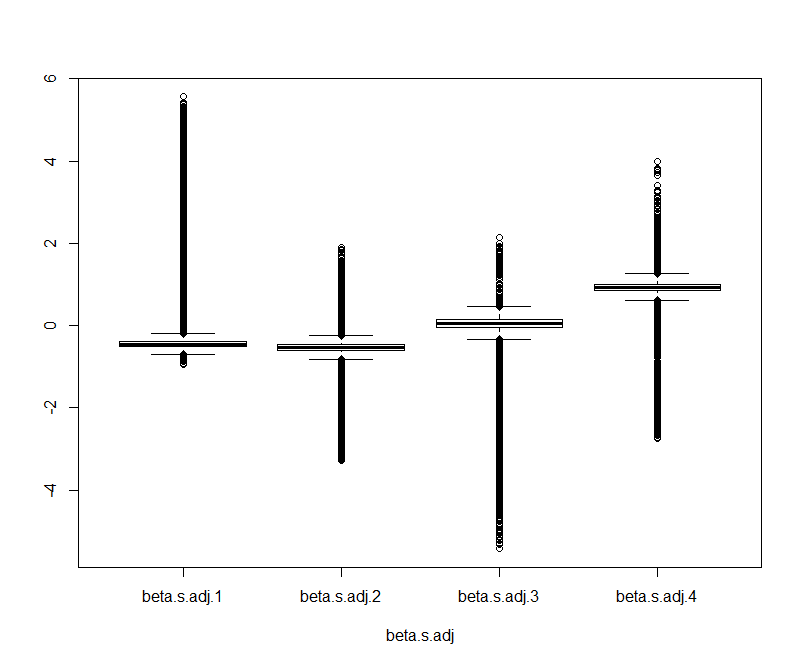


#The above plots show the convergence in the chains after 8000 iterations

#boxplots for smoking categories

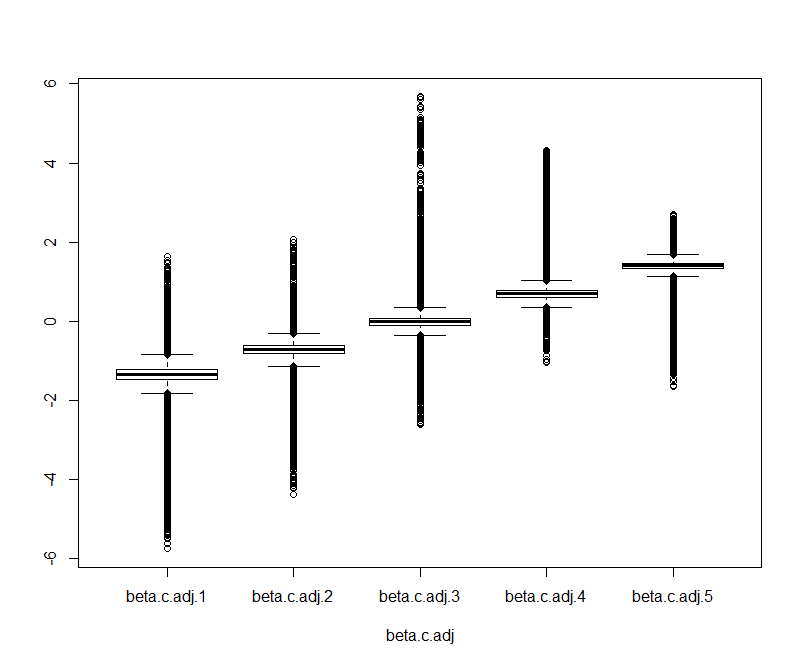
boxplot(data.frame( (smokeBug0$sims.list)["beta.s.adj"]),

xlab="beta.s.adj")

#boxplots for age categories

boxplot(data.frame( (smokeBug0$sims.list)["beta.c.adj"]),

xlab="beta.c.adj")



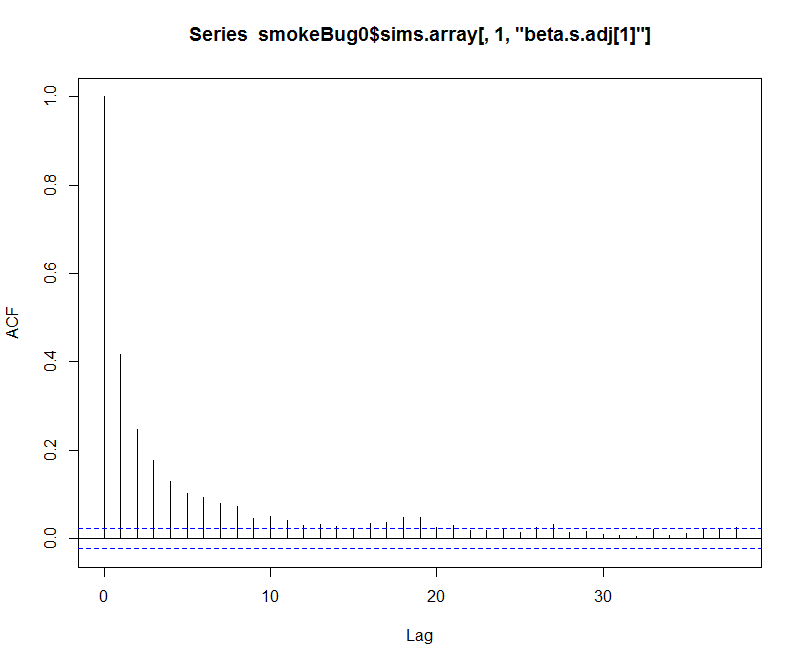
**#2b**

# the following plots show that our model converges

#autocorrelation functions given below

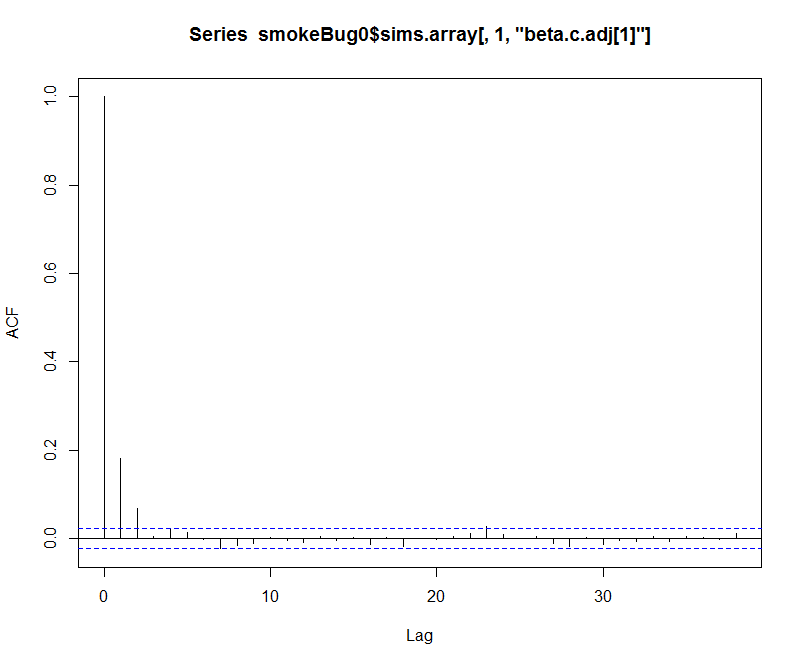
#trace plots have been shown in the output given above that shows the convergence in chains after 8000 iterations

acf(smokeBug0$sims.array[,1,"beta.s.adj[1]"])

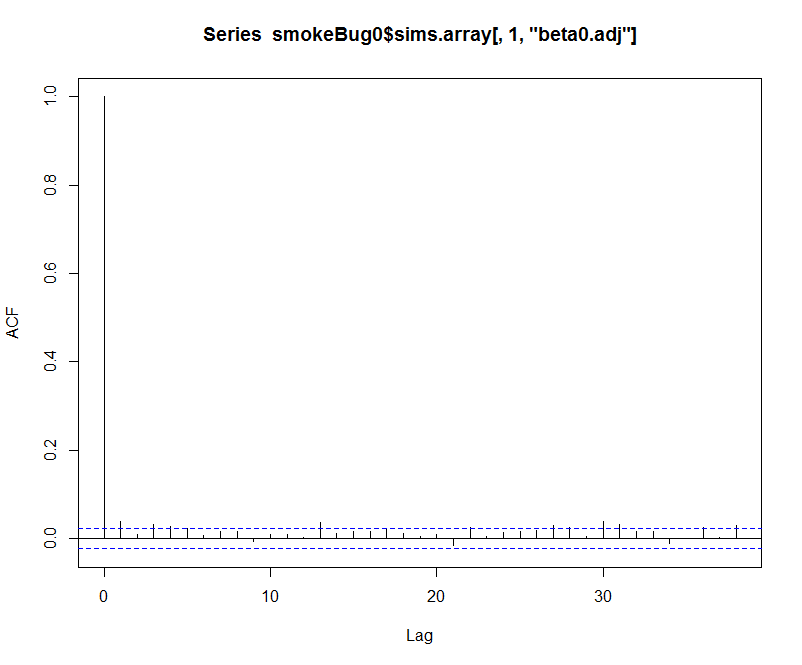


#After 10 lags the plot seems to have converged. We observe the same for age which converges twice as #fast as shown below

acf(smokeBug0$sims.array[,1,"beta.c.adj[1]"])



acf(smokeBug0$sims.array[,1,"beta0.adj"])



**#2c**

-R-code

> risk20<-smokeBug0$sims.list$beta.s.adj[,4]-smokeBug0$sims.list$beta.s.adj[,1]

> mean(risk20)

[1] 1.382443

> exp(mean(risk20))

[1] **3.984622**

**#The above shows those who smoke >20 cigarettes per day versus nonsmoker have approximately 4 times (=3.98) the risk. We take the exponential in the last line since the model gives us the log of the risks.**

> quantile(exp(risk20), c(.025, .975)) **# gives the 95% Cis for above risk**

2.5% 97.5%

2.763903 5.700207

**#Question3**

**#3a**

**#Done through R**

**#R-code given below for defining data parameters, model and priors**

rat.data <- read\_csv("C:/Users/omer\_hasan/Desktop/Current work/Bayesian Stats/assignment2/RatData5weeks.csv")

N<-50

y<-as.matrix(rat.data[,3:7])

y

week<-c(1,4,8,10,11)

attach(rat.data)

data<-list("N"=N, "y"=y,"dose"=dose,"week"=week)

cat("

model{

for(i in 1:N){

for(j in 1:5){

y[i,j]~dnorm(mu[i,j],tau)

mu[i,j] <- beta0[i]+beta1[i]\*week[j]

}

beta0[i]~dnorm(ma[i],ta)

beta1[i]~dnorm(mb[i],tb)

ma[i]<-beta00+ beta01\*(dose[i])

mb[i]<-beta10+ beta11\*(dose[i])

}

beta00~dnorm(100,.00001)

beta10~dnorm(0,.0001)

beta01~dnorm(0,.0001)

beta11~dnorm(0,.0001)

sau~dunif(0,250)

sa~dunif(0,250)

sb~dunif(0,250)

tau~dgamma(0.0001,0.0001)

ta~dgamma(0.0001,0.0001)

tb~dgamma(0.0001,0.0001)

}", file="ratmod3.txt")

paramsr=c("beta00","beta01","beta10", "beta11","tau")

init.funr=function(){list(

beta0=rnorm(50), beta1=rnorm(50),

beta00=rnorm(1), tau=rgamma(1,1,1),

beta10=rnorm(1), tb=rgamma(1,1,1),

beta01=rnorm(1), ta=rgamma(1,1,1),

beta11=rnorm(1))}

ratBug0=bugs(data, init.funr, paramsr, model.file="ratmod3.txt",

n.chains=5, n.iter=5000, n.burnin=500 , debug=TRUE #for production

# n.chains=5, n.iter=1000, n.burnin=100, n.thin=1, debug=TRUE #for testing

)

**#3b**

#showing autocorrelation plots to show convergence

#R-code

par(mfrow = c(2, 2))

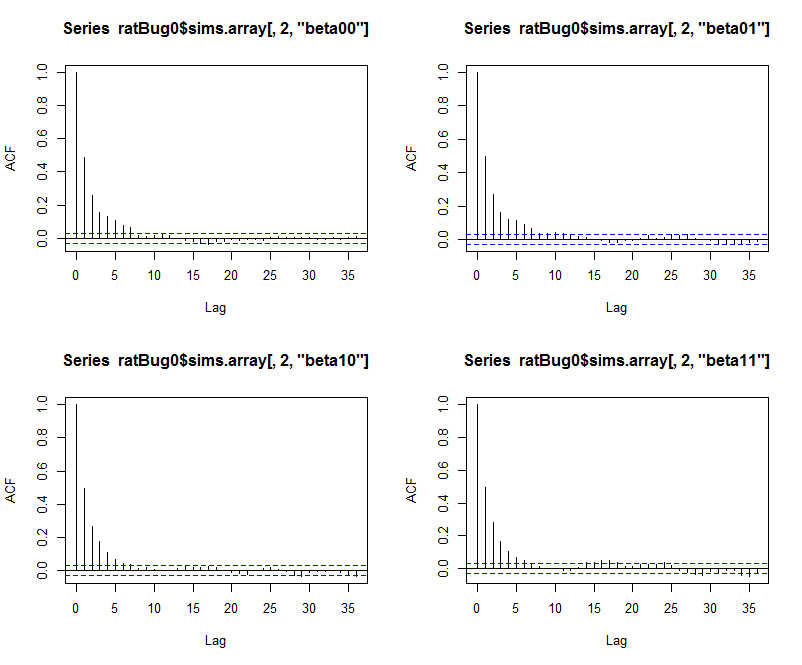
acf(ratBug0$sims.array[,2,"beta00"])

acf(ratBug0$sims.array[,2,"beta01"])

acf(ratBug0$sims.array[,2,"beta10"])

acf(ratBug0$sims.array[,2,"beta11"])

#autocorrelation Plots show the convergence for all at around 10 lags.



#We can also look at the trace plots to observe the convergence

**#3c**

**#output below shows posterior distributions of the betas and the precision.**

|  |
| --- |
| > print(ratBug0)  Inference for Bugs model at "ratmod3.txt",  Current: 5 chains, each with 5000 iterations (first 500 discarded)  Cumulative: n.sims = 22500 iterations saved  mean sd 2.5% 25% 50% 75% 97.5% Rhat n.eff  beta00 60.6 1.7 57.2 59.4 60.6 61.7 64.0 1 6900  beta01 -0.4 0.4 -1.3 -0.7 -0.4 -0.1 0.4 1 13000  beta10 7.1 0.2 6.7 6.9 7.1 7.2 7.4 1 9200  beta11 -0.2 0.0 -0.2 -0.2 -0.2 -0.1 -0.1 1 12000  tau 0.1 0.0 0.1 0.1 0.1 0.2 0.2 1 14000  deviance 1197.8 17.4 1166.0 1186.0 1197.0 1209.0 1235.0 1 22000  For each parameter, n.eff is a crude measure of effective sample size,  and Rhat is the potential scale reduction factor (at convergence, Rhat=1).  DIC info (using the rule, pD = Dbar-Dhat)  pD = 93.2 and DIC = 1291.0  DIC is an estimate of expected predictive error (lower deviance is better). |
|  |
| |  | | --- | | **#3d**  Yes there is a difference, the drug dose levels seem to **reduce** the growth of the rats compared to the ones that are not on the drug.  **Beta01 and Beta11** posterior distribution show the negative effect on growth. | |

**#3e**

#We can look at **Beta01 and Beta11** for dose levels.

Beta01, which has a gradient of -0.4, and standard deviation of 0.4, with a 95% CI of [-1.3, 0.4], we can see that there the drug dose level will slightly reduce the growth of the rats. Beta11 exhibits the same nature with a negative gradient, which is more precise and shows a negative relationship between dose levels over time and growth.

**Appendices**

**Appendix for Q1 part b**

#Importing data and defining initial variables

nbig<-20000

set.seed(25)

question1data <- read\_csv("C:/Users/omer\_hasan/Desktop/Current work/Bayesian Stats/assignment2/question1data.csv")

Y1<- mean(question1data$Y1)

Y1

Y2<- mean(question1data$Y2)

Y3<- mean(question1data$Y3)

Y4<- mean(question1data$Y4)

temp40<- question1data$Y1

Y1

temp60<- question1data$Y2

temp80<- question1data$Y3

temp100<- question1data$Y4

#Algorithm used

if(T){

mu00<-0

tau00<-1

alp<-1

beta<-1

alp0<-1

beta0<-1

mu1<-rep(0,nbig)

mu2<-rep(0,nbig)

mu3<-rep(0,nbig)

mu4<-rep(0,nbig)

tau<-rep(0,nbig)

mu0post1<-rep(0,nbig)

tau0post1<-rep(0,nbig)

mu0post2<-rep(0,nbig)

tau0post2<-rep(0,nbig)

mu0post3<-rep(0,nbig)

tau0post3<-rep(0,nbig)

mu0post4<-rep(0,nbig)

tau0post4<-rep(0,nbig)

mu0<-rep(0,nbig)

tau0<-rep(0,nbig

mupost00<-rep(0,nbig)

taupost00<-rep(0,nbig)

betapost<-rep(0,nbig)

betapost0<-rep(0,nbig)

mu0[1]<-1

mu1[1]<-0

mu2[1]<-0

mu3[1]<-0

mu4[1]<-0

tau[1]<-1

tau0[1]<-1

for(i in 2:nbig){

mu0post1[i]<-(tau0[i-1]\*mu0[i-1]+5\*tau[i-1]\*Y1)/(tau0[i-1]+5\*tau[i-1])

tau0post1[i]<-tau0[i-1]+tau[i-1]\*5

mu1[i]<-rnorm(1,mu0post1[i],sd=1/sqrt(tau0post1[i]))

mu0post2[i]<-(tau0[i-1]\*mu0[i-1]+5\*tau[i-1]\*Y2)/(tau0[i-1]+5\*tau[i-1]

tau0post2[i]<-tau0[i-1]+tau[i-1]\*5

mu2[i]<-rnorm(1,mu0post2[i],sd=1/sqrt(tau0post2[i]))

mu0post3[i]<-(tau0[i-1]\*mu0[i-1]+5\*tau[i-1]\*Y3)/(tau0[i-1]+5\*tau[i-1])

tau0post3[i]<-tau0[i-1]+tau[i-1]\*5

mu3[i]<-rnorm(1,mu0post3[i],sd=1/sqrt(tau0post3[i]))

mu0post4[i]<-(tau0[i-1]\*mu0[i-1]+5\*tau[i-1]\*Y4)/(tau0[i-1]+5\*tau[i-1])

tau0post4[i]<-tau0[i-1]+tau[i-1]\*5

mu4[i]<-rnorm(1,mu0post4[i],sd=1/sqrt(tau0post4[i]))

mupost00[i]<-(4\*tau0[i-1]\*((mu1[i-1]+mu2[i-1]+mu3[i-1]+mu4[i-1])/4)+tau00\*mu00)/(4\*tau0[i-1]+tau00)

taupost00[i]<-4\*tau0[i-1]+tau00

mu0[i]<-rnorm(1,mupost00[i],1/sqrt(taupost00[i]))

alppost0<-alp0+(4/2)

betapost0[i]<-beta0+(1/2)\*(sum((mu1[i]-mu0[i])^2)+sum((mu2[i]-mu0[i])^2)+sum((mu3[i]-mu0[i])^2)+sum((mu4[i]-mu0[i])^2))

tau0[i]<-rgamma(1,alppost0,betapost0[i])

alppost<-alp+(5+5+5+5)/2

betapost[i]<- beta+(0.5\*(sum((temp40-mu1[i])^2)+sum((temp60-mu2[i])^2)+sum((temp80-mu3[i])^2)+sum((temp100-mu4[i])^2)))

tau[i]<-rgamma(1,alppost,betapost[i])

}

}

#R-code

> cbind(mu0,mu1,mu2,mu3,mu4,tau,tau0)[1:10,]

#output

mu0 mu1 mu2 mu3 mu4 tau tau0

[1,] 1.0000000 0.0000000 0.000000 0.000000 0.000000 1.000000 1.0000000

[2,] -0.5925370 1.3170689 1.665680 2.059699 2.105177 2.515214 0.1363786

[3,] 0.3722076 0.9116370 1.616226 1.912114 3.071285 6.827713 0.1490493

[4,] 1.4159503 1.1532206 2.148372 2.020827 2.979335 3.177054 0.9635967

[5,] 2.2172000 0.8751746 1.555908 2.361955 3.321319 5.150798 0.5602225

[6,] 0.5317665 1.1355463 1.960274 2.122645 3.081795 6.101424 0.5607278

[7,] 1.0260886 1.1105413 1.779044 1.847387 3.195380 6.894374 0.6908207

[8,] 1.0391479 0.9005278 1.673157 2.620379 2.963925 7.859723 0.2967751

[9,] 1.7151787 1.0002248 1.622928 1.924740 3.340461 7.409448 1.1356559

[10,] 2.7154530 1.3106049 1.718507 2.407333 3.125839 6.335775 0.6744737